

000007-2008260

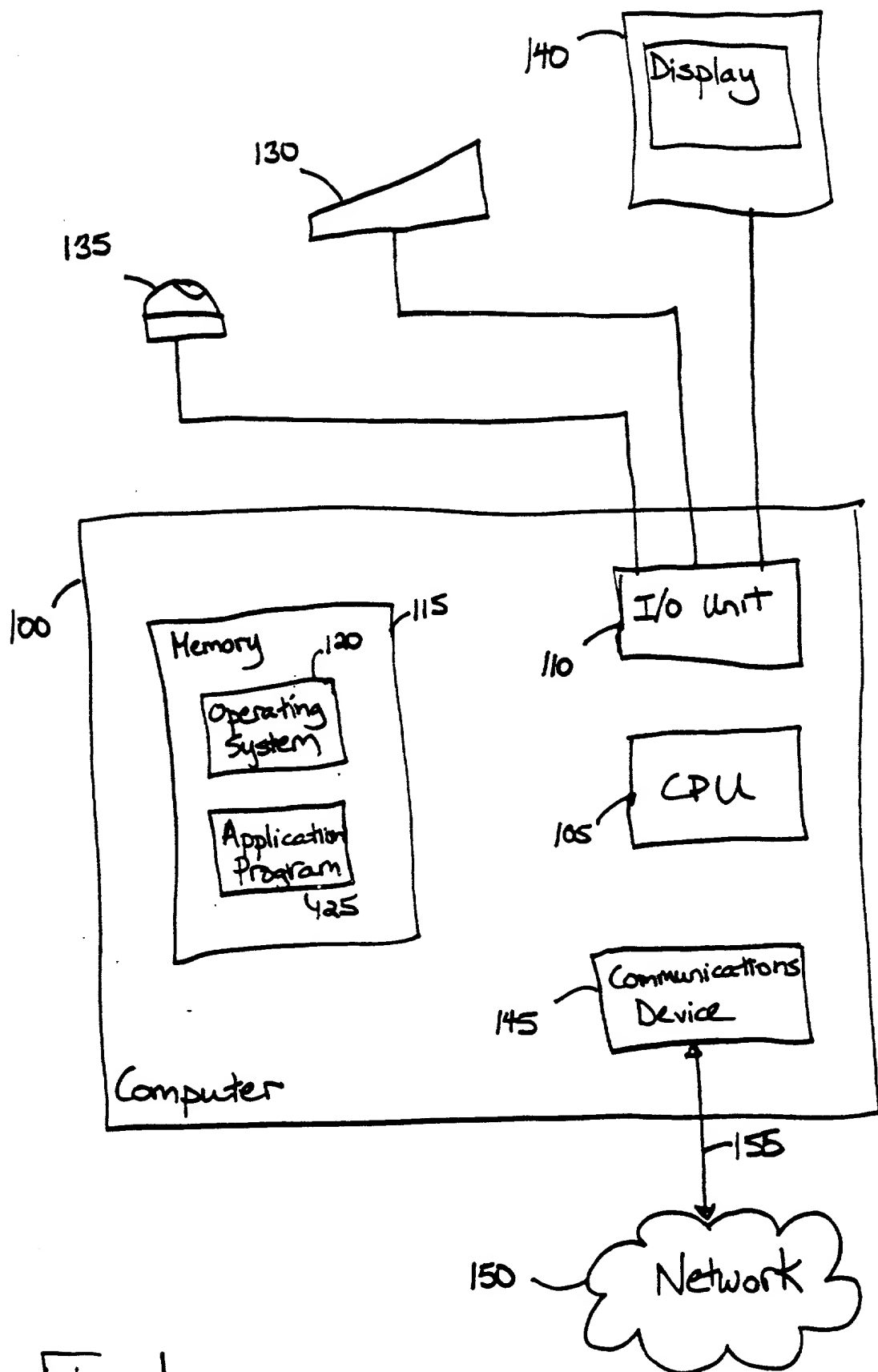


Fig. 1

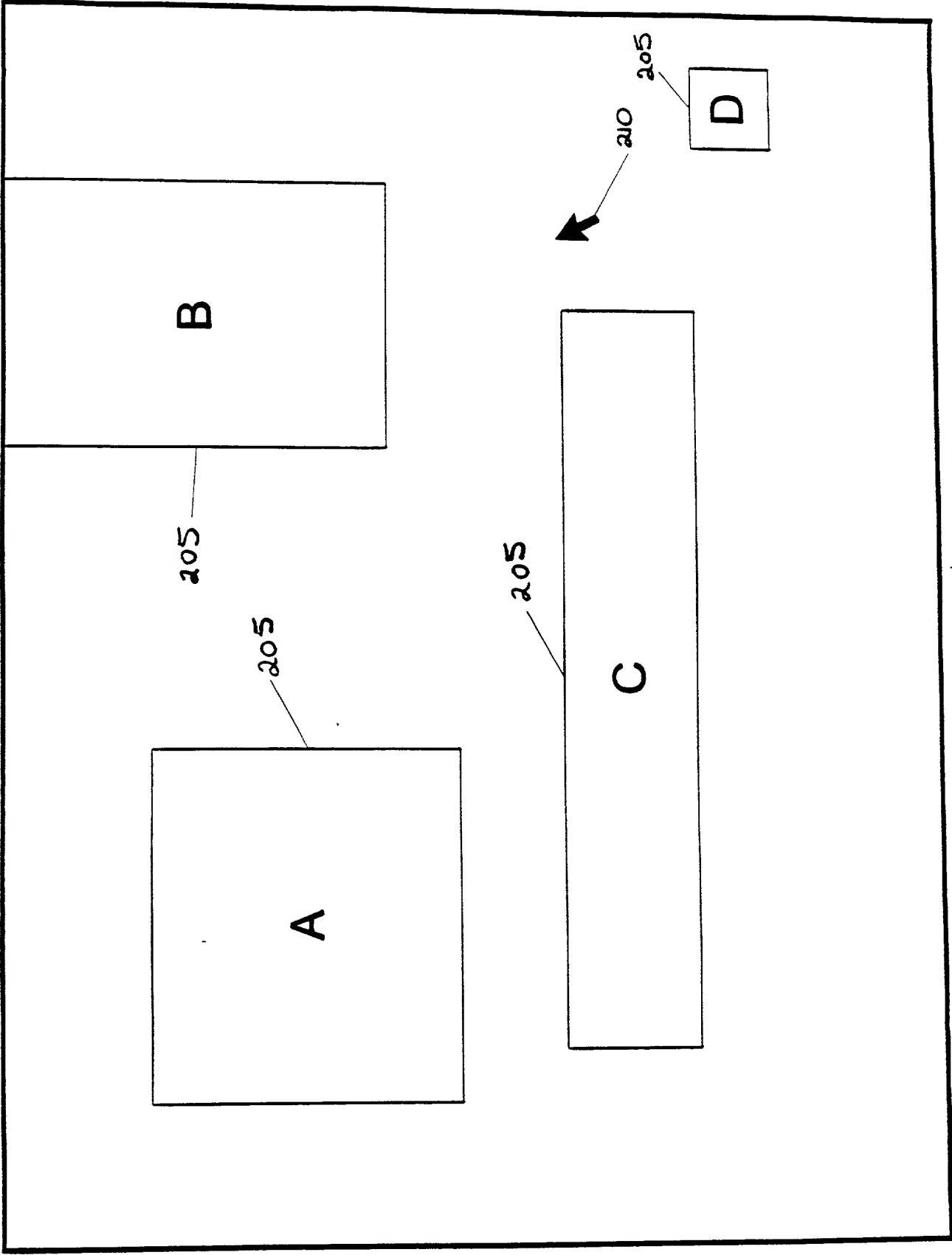


Fig. 2
PRIOR ART

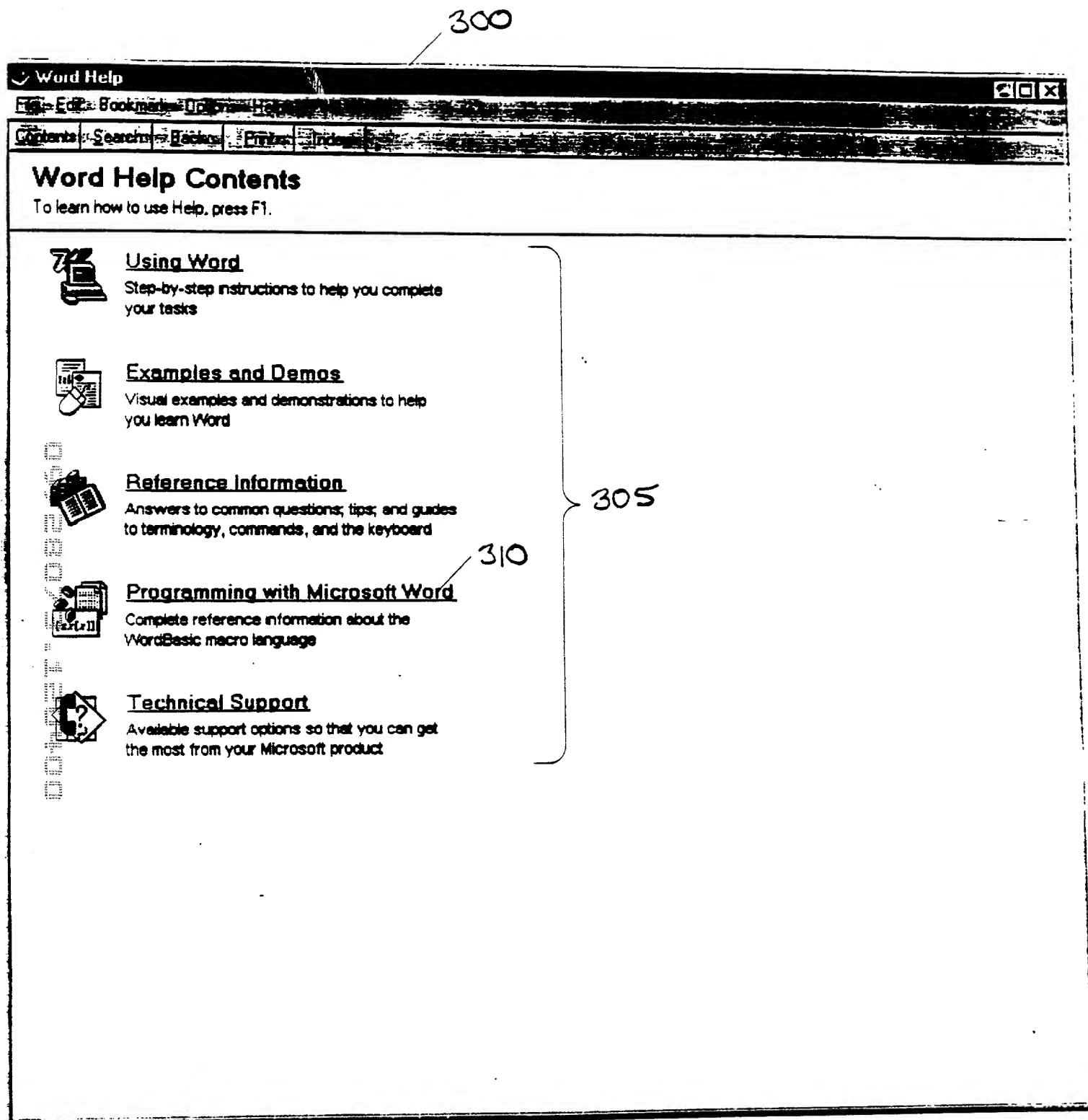


Fig. 3A
PRIOR ART

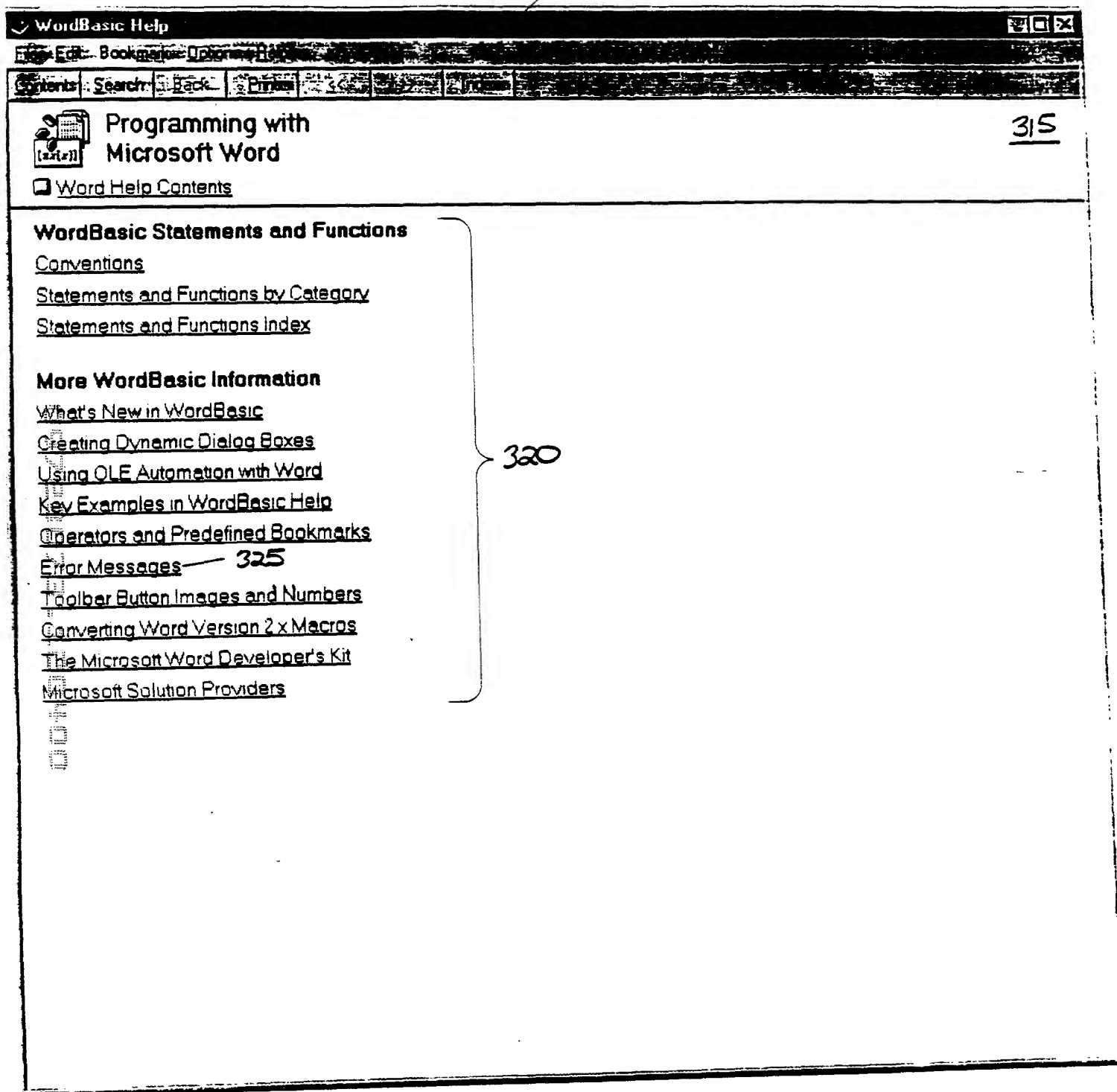


Fig. 3B
PRIOR ART

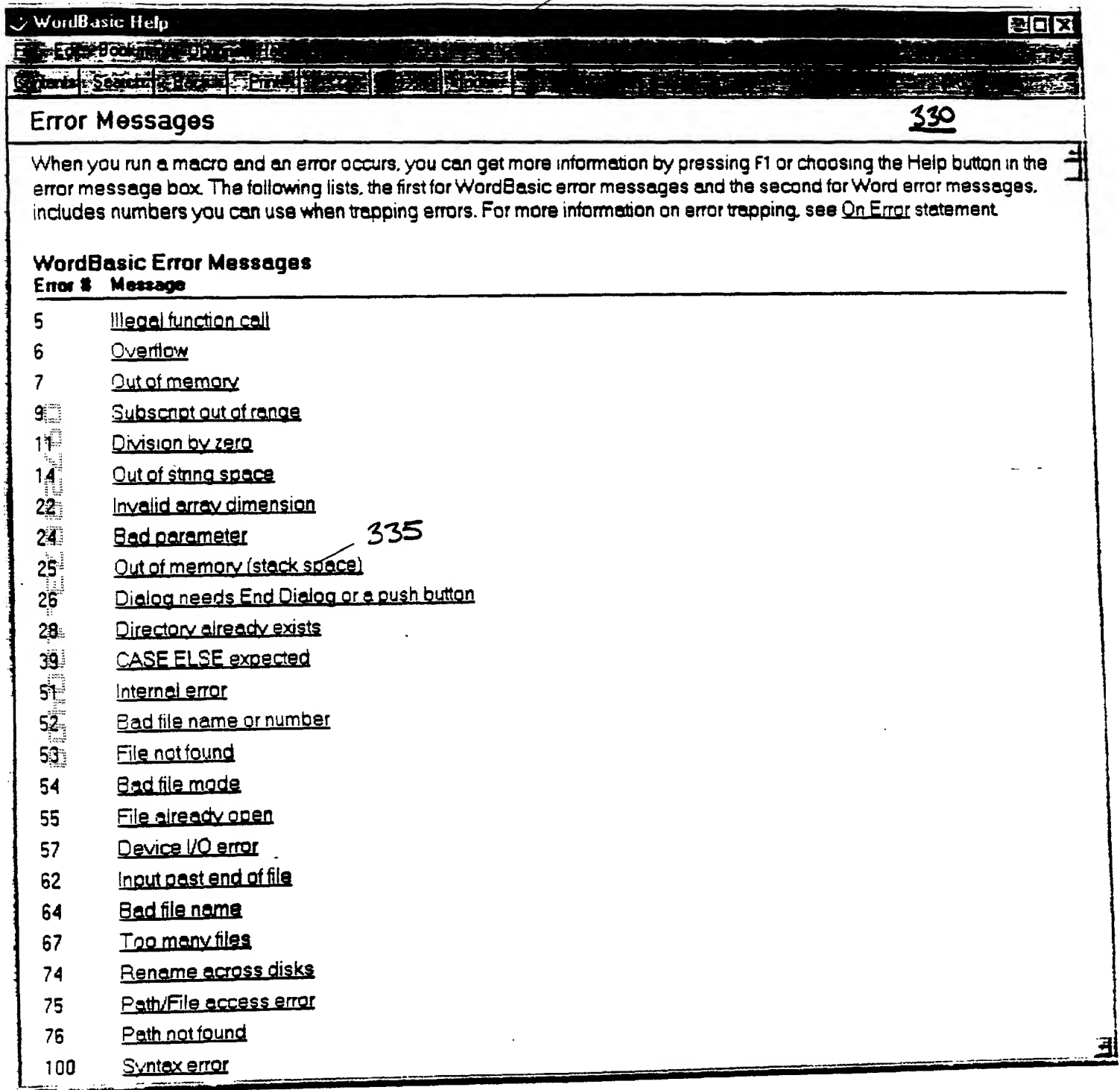


Fig. 3C
PRIOR ART

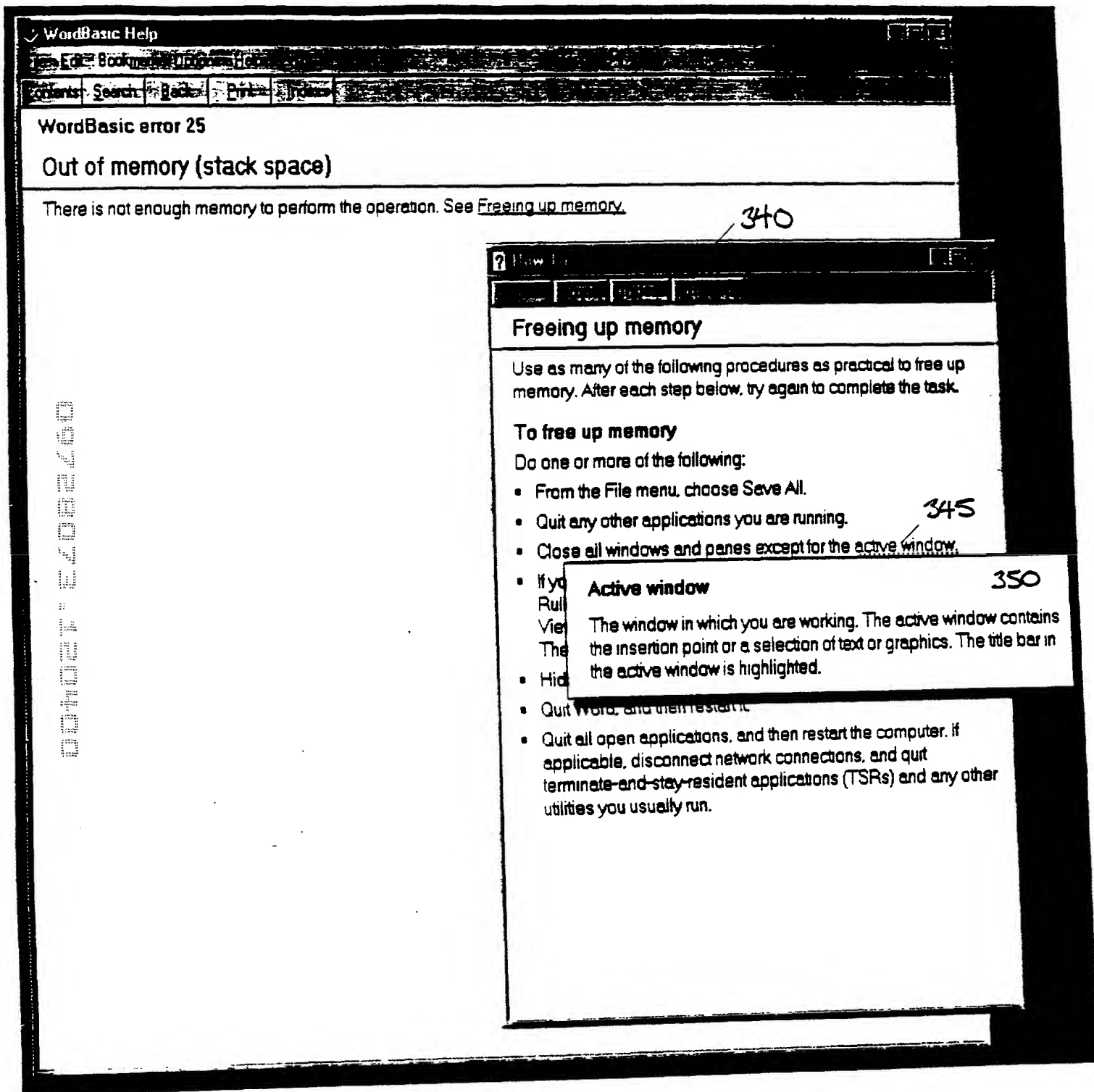


Fig. 3D
PRIOR ART

410

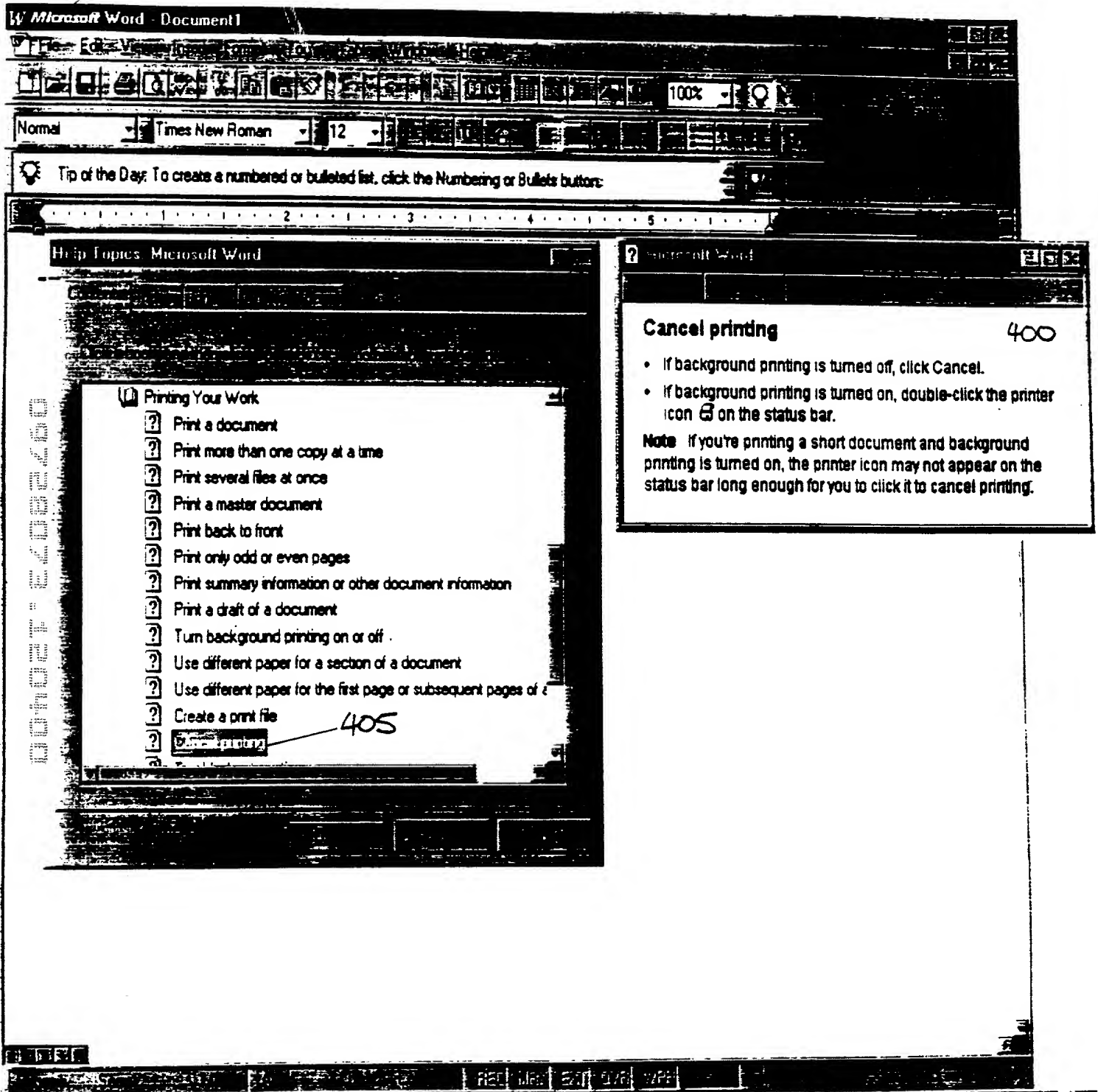


Fig. 4
PRIOR ART

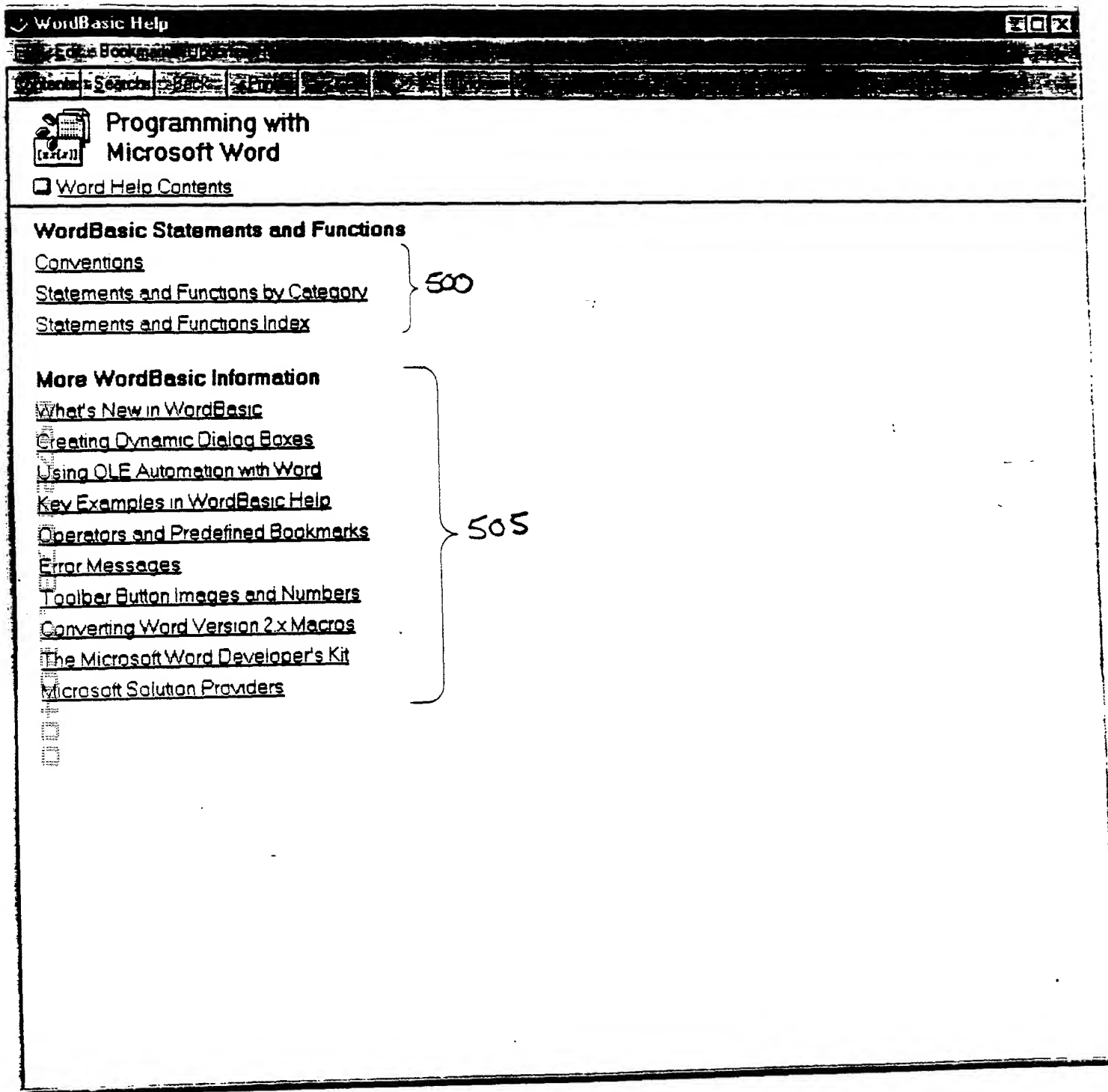


Fig. 5A
PRIOR ART

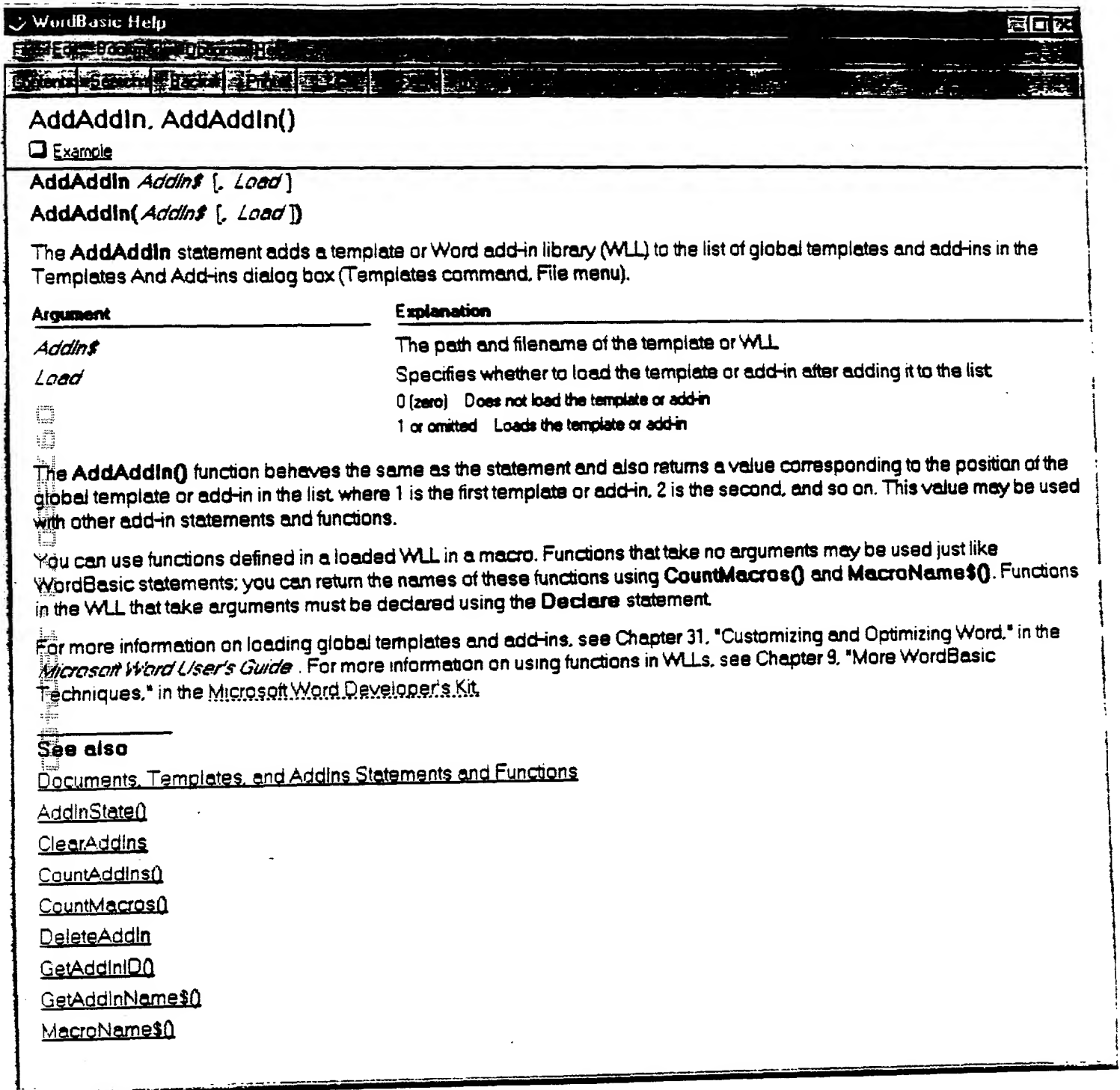


Fig. 5B
PRIOR ART

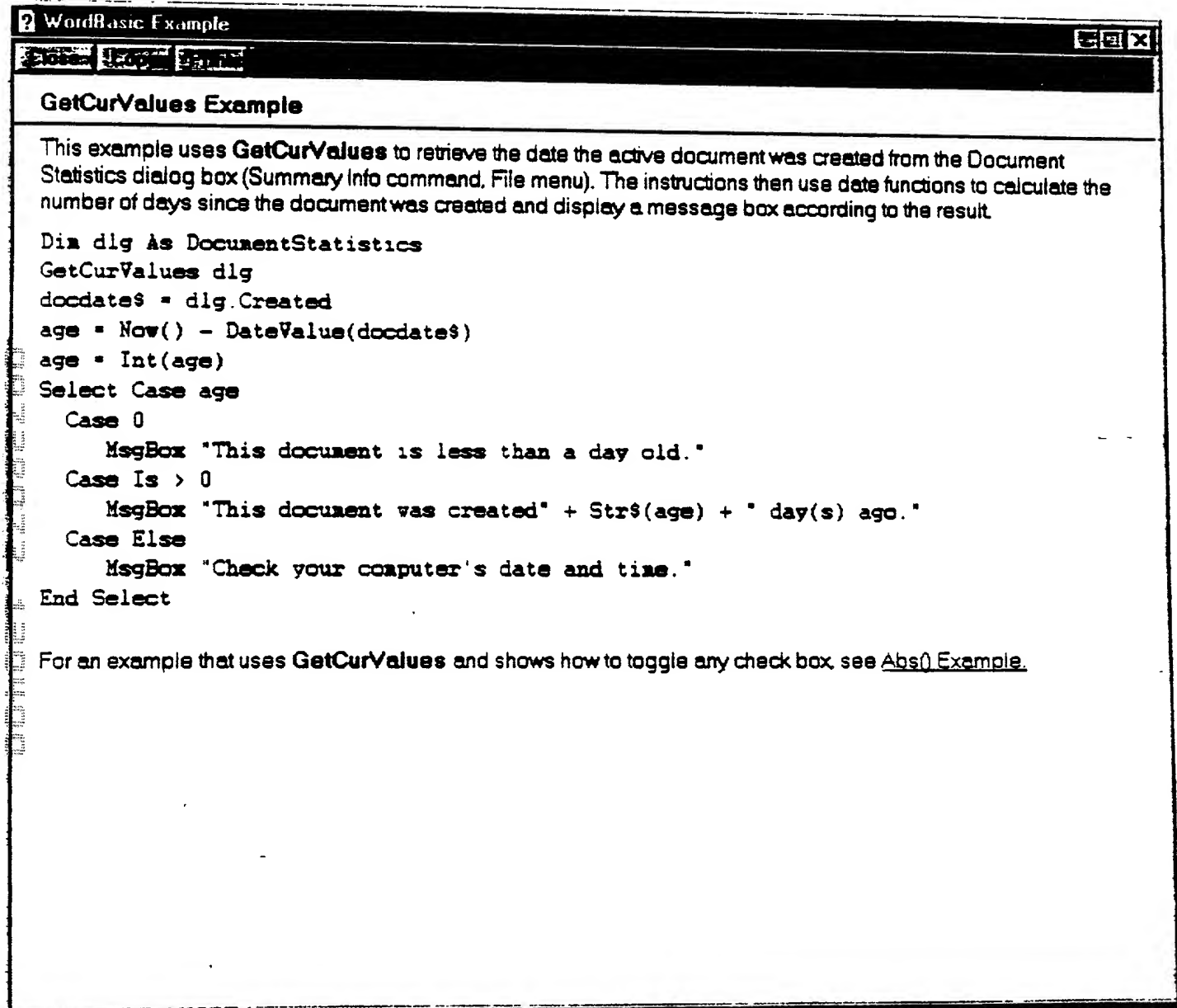


Fig. 5C
PRIOR ART

610

http://www.aip.org/ Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites History Mail Print Edit

Address http://www.aip.org/ Go

AMERICAN INSTITUTE OF PHYSICS • What's New • Help/Feedback • Search

AIP and its Member Societies

Public Information

Online Journal Publishing Service

AIP Journals

Magazines, Books & Proceedings

Publishing Services

Employment & Industry

Education & Student Services

Science Policy

History Center

IN THE SPOTLIGHT

Physics Societies Announce Joint Venture to Launch Virtual Journals

The American Institute of Physics (AIP) and the American Physical Society (APS) announced that the first two of a series of "virtual" journals in the physical sciences are set to launch in January, 2000.

Press Release

APS Division of Plasma Physics Employment Center

Physics and Chemistry Nobel Prizes

Two Dutch physicists win for their work toward deriving a unified framework for the forces in nature; a Caltech chemist wins for developing an ultrafast camera that captures all the steps of a chemical reaction. More detail in Physics News Update.

Find out more...

625

615

625

620

600

605

Fig. 6

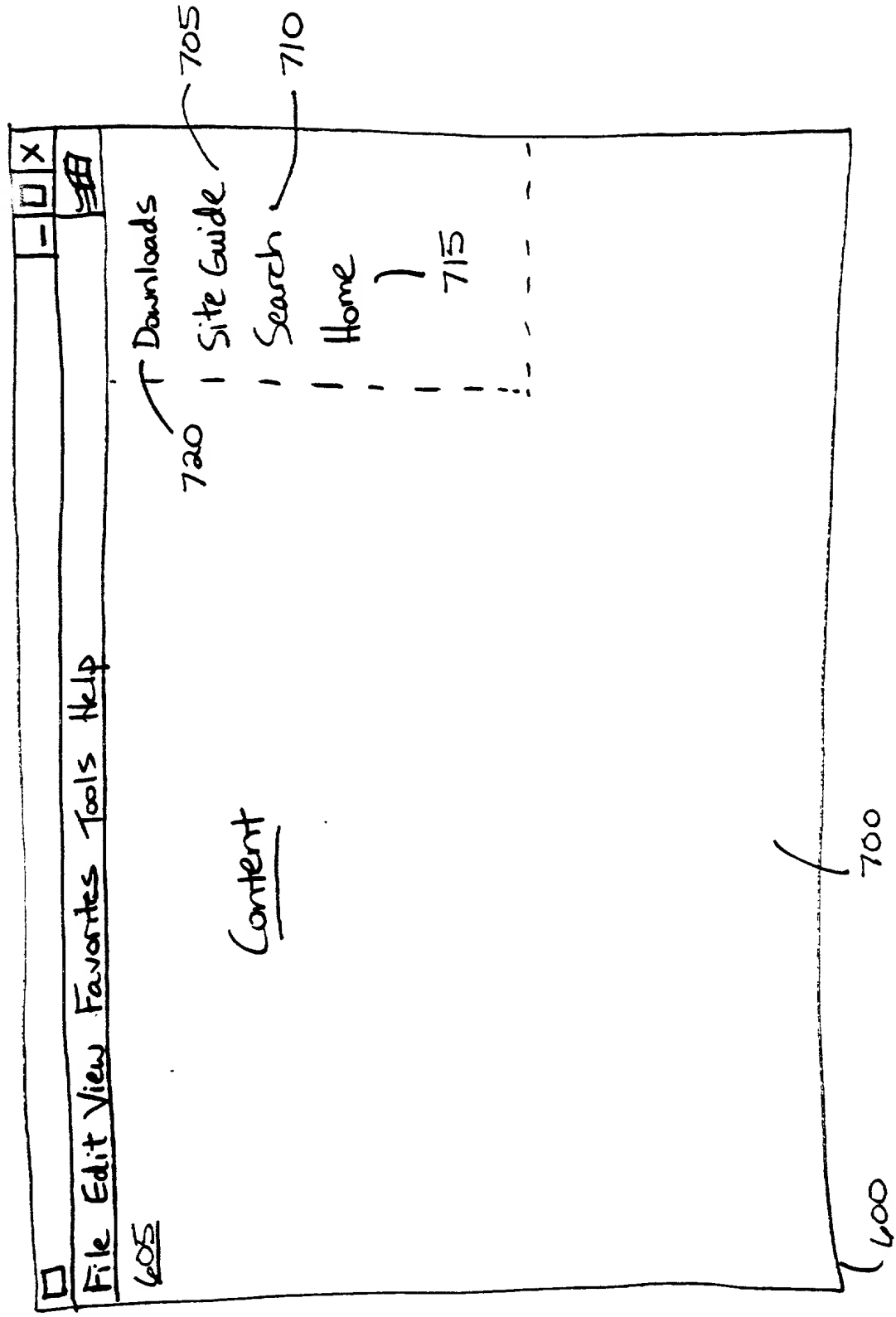


Fig. 7

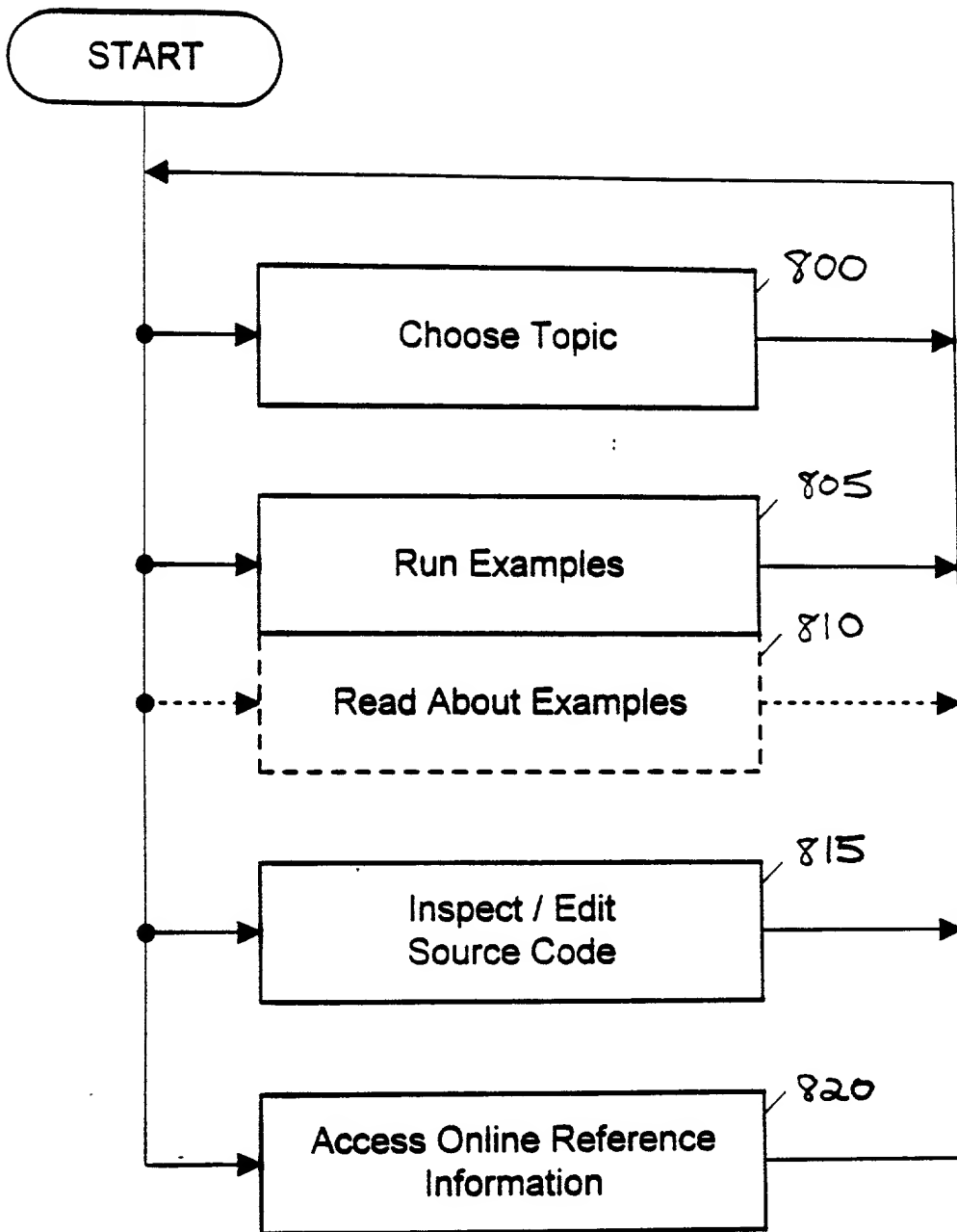


Fig. 8

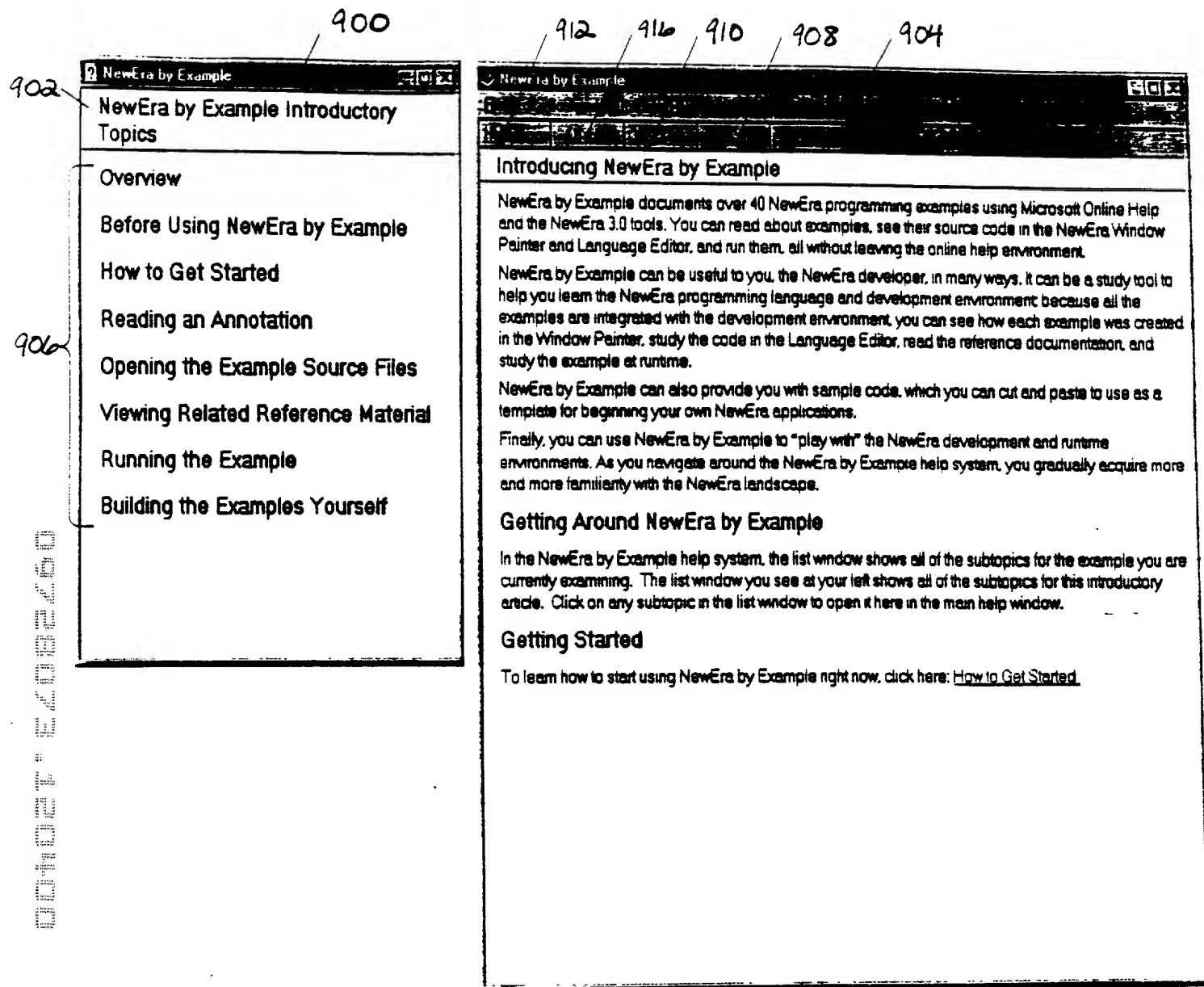


Fig. 9A

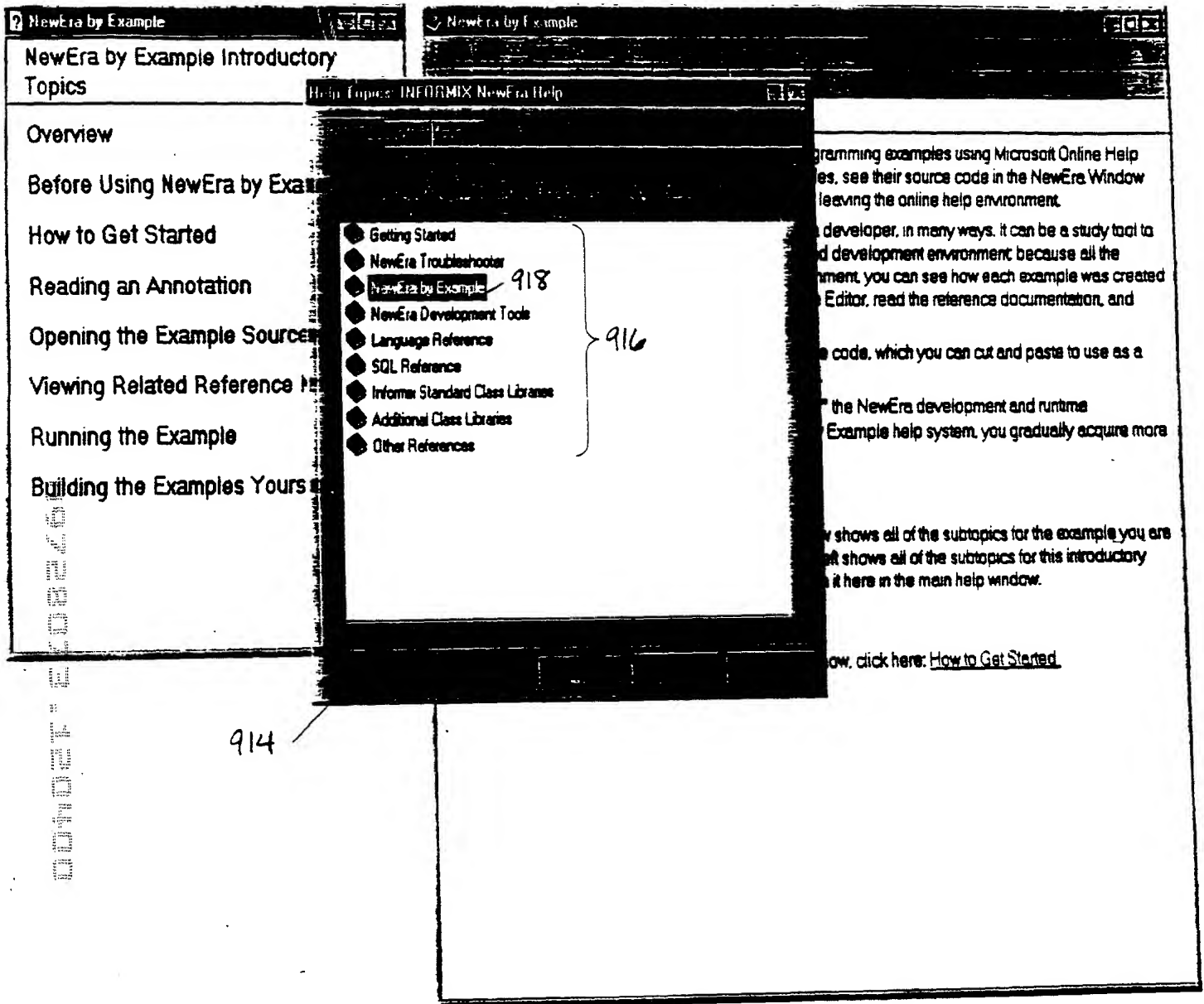


Fig. 9B

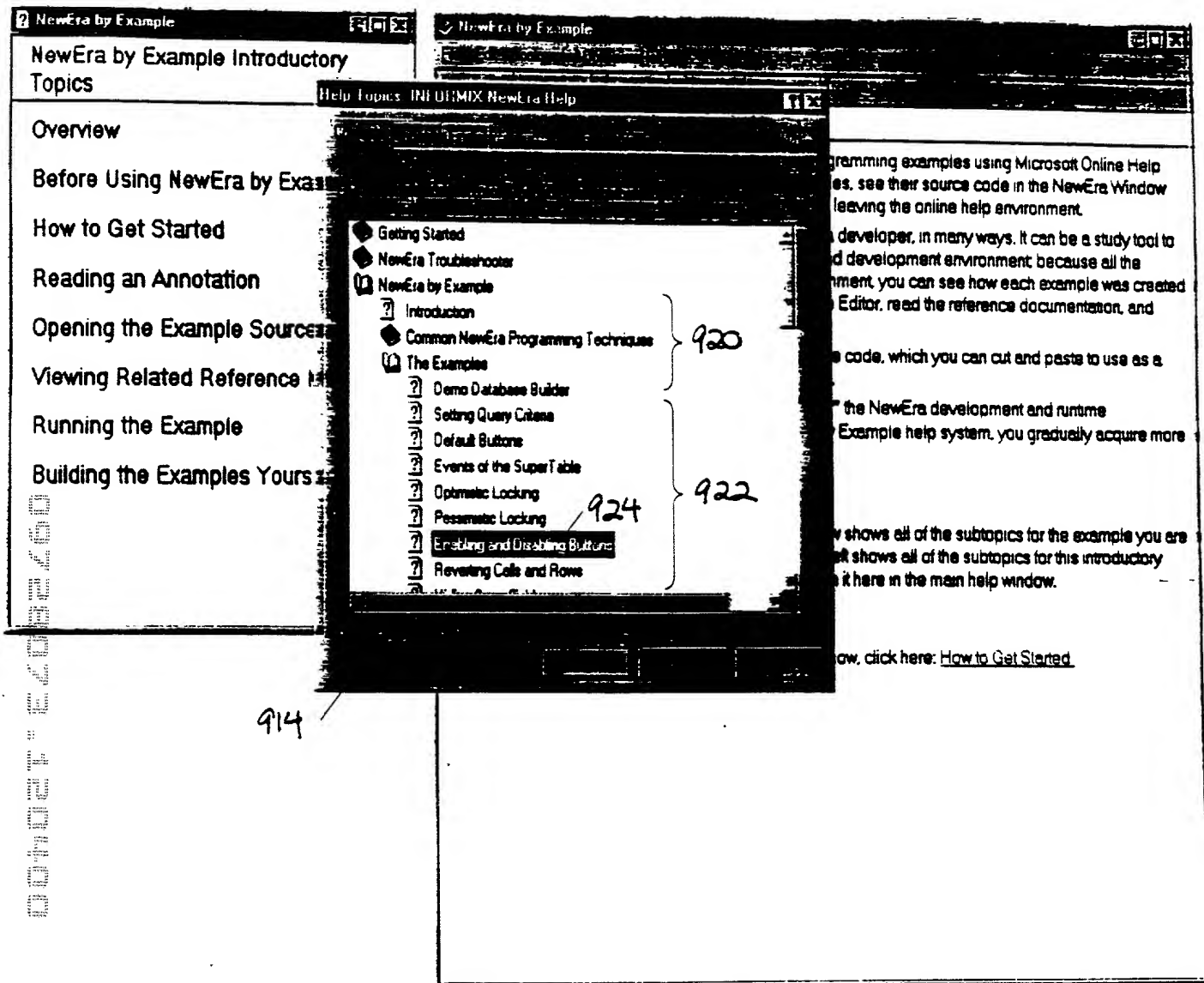


Fig. 9C

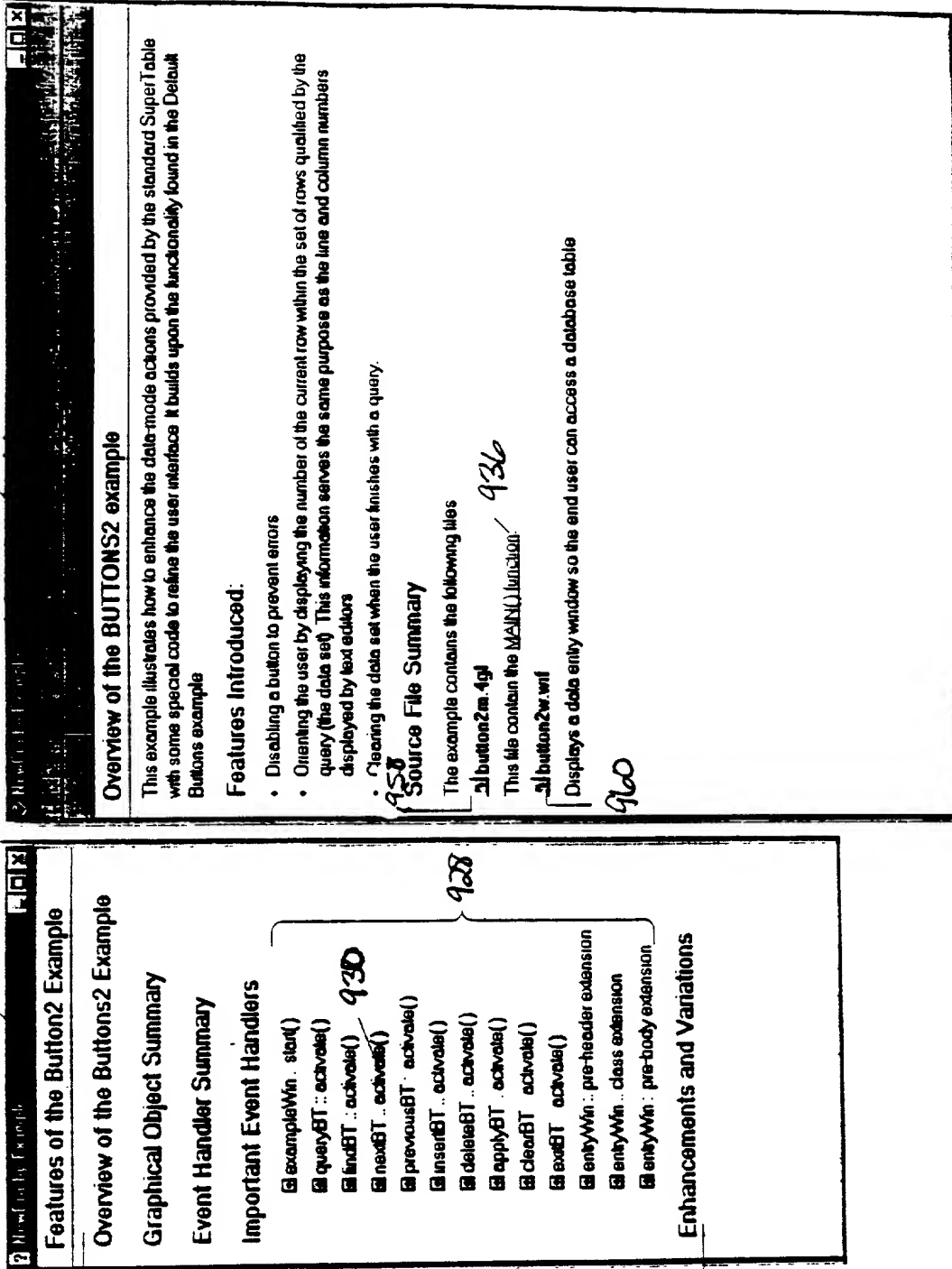


Fig. 9D

958

The MAIN() FUNCTION

button2m.4gl:

MAIN

944 VARIABLE exampleWindow exampleWin

CALL ixSQLConnect::getImplicitConnection().connect("Sports")

942 LET exampleWindow = NEW exampleWin()

CALL exampleWindow.open()

RETURN

END MAIN

940

Fig. 9F

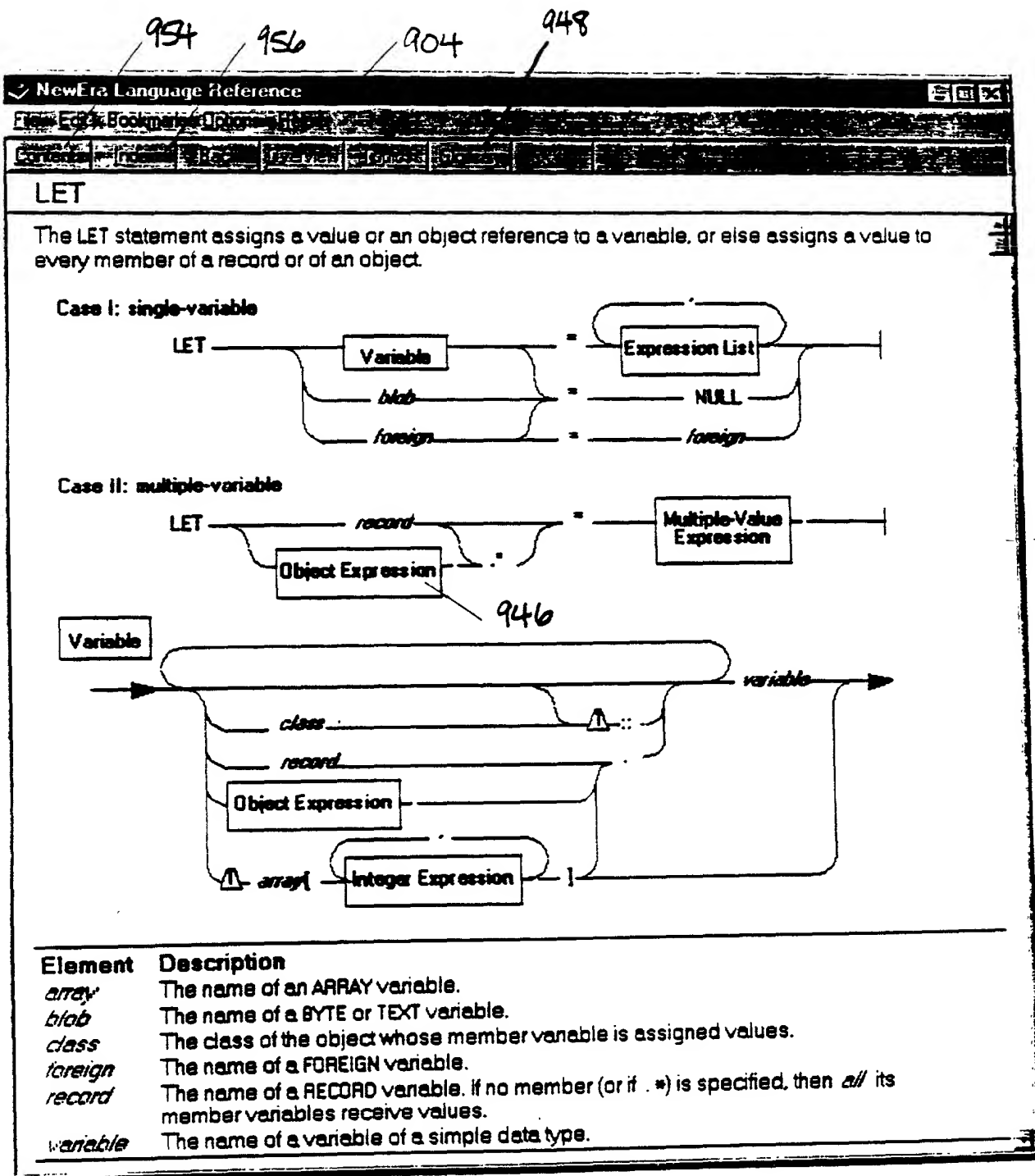


Fig. 9G

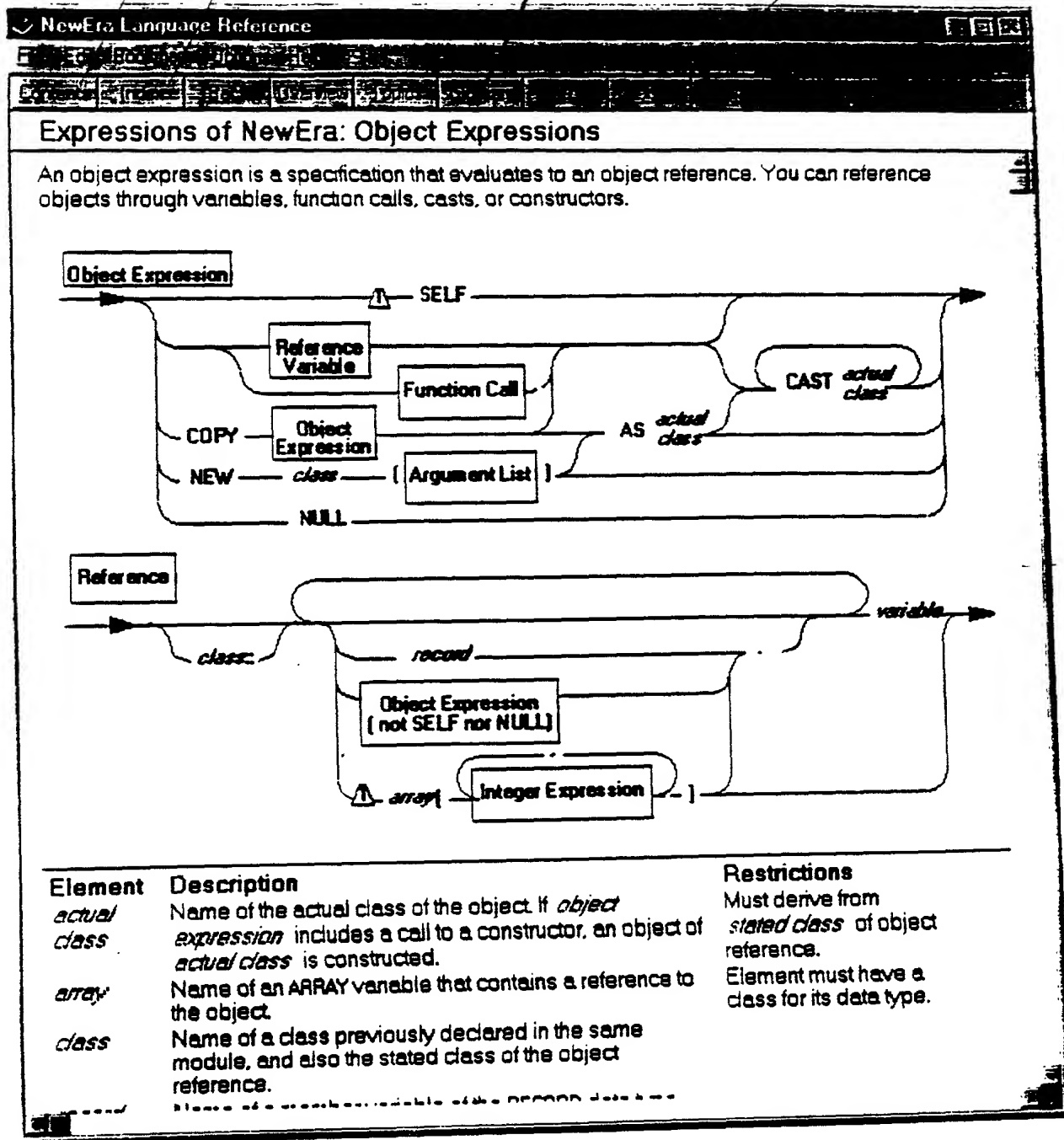


Fig. 9H

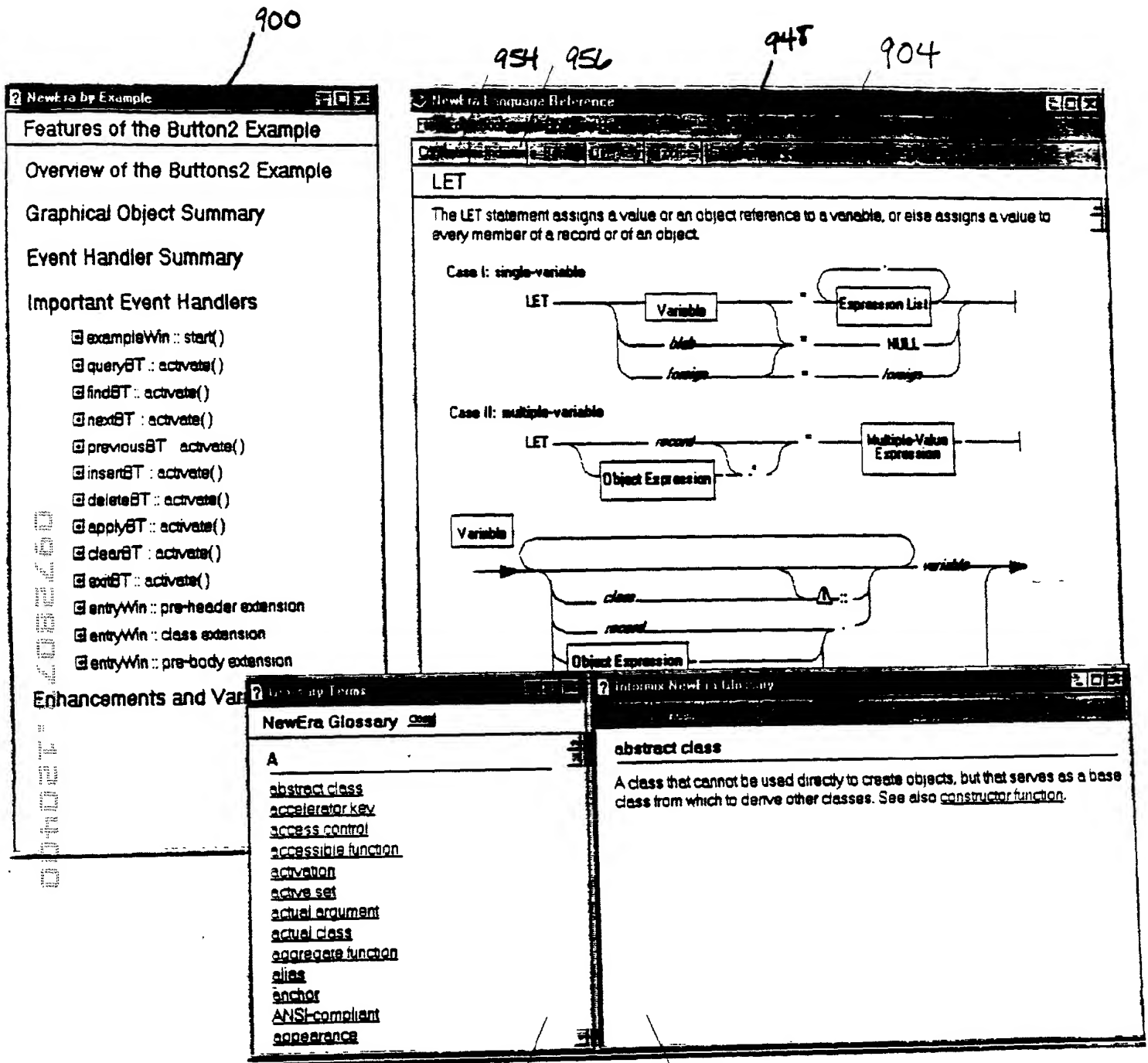


Fig. 9I

Codewright for NewEra

File Edit Search Project Window Help

Code

C:\NewEra3\0\examples\buttons2\button2m.4gl

```
INCLUDE "annotate.4gh"
INCLUDE SYSTEM "ixconn.4gh"
INCLUDE "button2w.4gh"

MAIN
  VARIABLE exampleWindow exampleWin
  CALL nebyex::annotate(buttons2_MAIN)
  CALL ixSQLConnect::getImplicitConnection().connect("Sports")
  LET exampleWindow = NEW exampleWin()
  CALL exampleWindow.open()

  RETURN

END MAIN
```

Line 5 Col 1

NewEra by Example

Fig. 4J

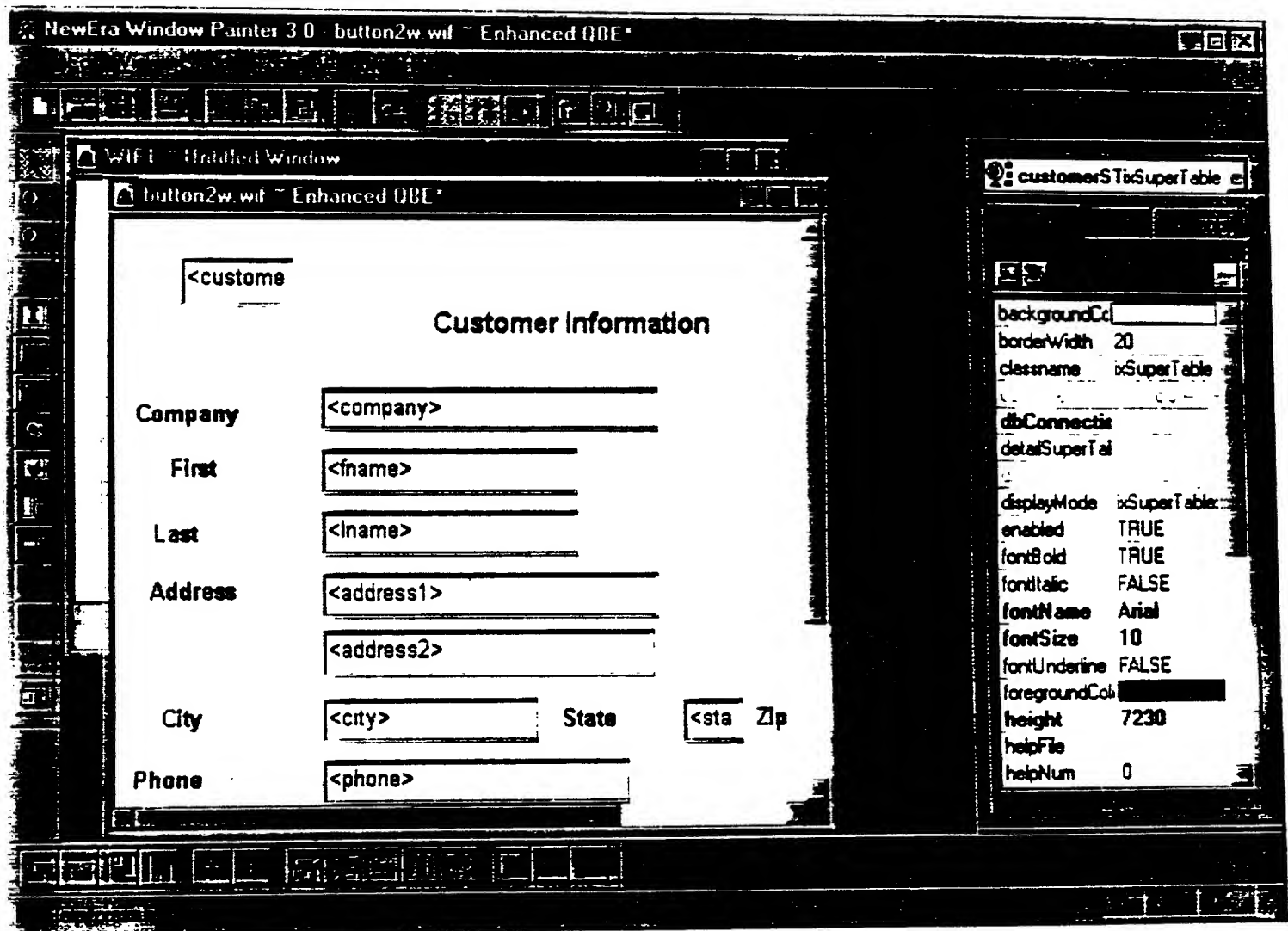


Fig. 4K

900

NewEra by Example
Features of the Edit Menu Example
Overview of the Edit Menu Example
The MAIN () function
Graphical Object Summary
Event Handler Summary
Important Event Handlers
<input type="checkbox"/> edit1TB :: focusin () <input type="checkbox"/> edit2TB :: focusin () <input type="checkbox"/> noneditCB :: focusin () <input type="checkbox"/> editMI :: activate () <input type="checkbox"/> cutMI :: activate () <input type="checkbox"/> copyMI :: activate () <input type="checkbox"/> pasteMI :: activate () <input type="checkbox"/> deleteMI :: activate ()
Extension Summary
Important Extensions
<input type="checkbox"/> editWin :: class extension <input type="checkbox"/> editWin :: preheader extension <input type="checkbox"/> editWin :: prebody extension

904 962

NewEra by Example
Overview of Edit Menu
This example provides a standard Edit menu to cut, copy, paste, and delete text from a text box.
Features Introduced
<ul style="list-style-type: none"> • Creating menus and menu items. • Executing code when the user chooses a menu item. • Executing built-in clipboard functions. • Finding the current control in an application. • Specifying an accelerator key for a menu item. • Executing code when the user enters a control. • Enabling and disabling menu items.
Source File Summary
<input type="checkbox"/> editm.4gl Initializes the application by creating the editWin window. The MAIN () function of the application is in this file.
<input type="checkbox"/> editw.wtl Provides a window with an edit menu and some demonstration text boxes.

Fig. 9L

900

Newtra by Example

Features of the Edit Menu Example

Overview of the Edit Menu Example

The MAIN () function

Graphical Object Summary

Event Handler Summary

Important Event Handlers

- edit1TB :: focusin ()
- edit2TB :: focusin ()
- noneditCB :: focusin ()
- extMI :: activate ()
- cutMI :: activate ()
- copyMI :: activate ()
- pasteMI :: activate ()
- deleteMI :: activate ()

Extension Summary

Important Extensions

- editWin :: class extension
- editWin :: preheader extension
- editWin :: prebody extension

904

Newtra by Example

The MAIN () Function

editm.4gl:

```

MAIN
VARIABLE
    editWN editWin
    LET editWN = NEW editWin()
    CALL editWN.open()
    RETURN
END MAIN

```

Initializes the application by creating the editWin window. MAIN defines a variable in which to store a reference to the created window, creates the window, and then opens the window.

In fact, the same actions could be performed without defining a variable as in the following example:

```

MAIN
    CALL ( NEW editWin ( ) ).open ( )
    RETURN
END MAIN

```

The trick here is that we only need a reference to the editWin window to qualify the call to open (). Once the window is opened, the window can take care of itself. Thus, instead of capturing the reference generated by NEW in a variable, we use it instead to qualify the call to open ().

904

Edit Window

Text to edit

More text to edit

CheckBox

Use the edit menu to cut and paste text from one textbox to the other. Use the button to start a text editor so you can paste in the editor. Tab to the checkbox and button to see disabling of the menu items.

Fig. 9M

900

NewEra by Example
Features of the Edit Menu Example
Overview of the Edit Menu Example
The MAIN () function
Graphical Object Summary
Event Handler Summary
Important Event Handlers
<input type="checkbox"/> edit1TB :: focusin () <input type="checkbox"/> edit2TB :: focusin () <input type="checkbox"/> noneEditCB :: focusin () <input type="checkbox"/> editM1 :: activate () <input type="checkbox"/> editM2 :: activate () <input type="checkbox"/> copyM1 :: activate () <input type="checkbox"/> pasteM1 :: activate () <input type="checkbox"/> deleteM1 :: activate ()
Extension Summary
Important Extensions
<input type="checkbox"/> editWin :: class extension <input type="checkbox"/> editWin :: preheader extension <input type="checkbox"/> editWin :: prebody extension

904

NewEra by Example
edit1TB :: focusin ()
editw.wm - in edit1TB handler for xTextBox:focusin event
VARIABLE win editWin = getWindow () CALL win.setEditItemsEnabled (SELF)
LET editWin = getWindow () In the handler of the window, the members of the window are in scope. The members in scope include the graphical objects that you paint within the window. In the handlers of other graphical objects, however, the members of the window are not in scope. You have to qualify a member with a reference to the window. Each graphical object has the getWindow () member function, which conveniently returns a reference to the window. You can capture the reference in a local variable of the handler. CALL win.setEditItemsEnabled (TRUE) The example uses the reference to qualify the call to the setEditItemsEnabled () function. The call passes the TRUE parameter to enable the editing menu items while the user is in the text box.

904

Text Window
<div style="display: flex; justify-content: space-between;"> <div> <p>Text to edit 906</p> <p>More text to edit 908</p> </div> <div> <p>910</p> <p>CheckBox</p> </div> </div>
Use the edit menu to cut and paste text from one textbox to the other. Use the button to start a text editor so you can paste in the editor. Tab to the checkbox and button to see disabling of the menu items.

Fig. 9N

NewEra by Example	
Features of the Edit Menu Example	
Overview of the Edit Menu Example	
The MAIN () function	
Graphical Object Summary	
Event Handler Summary	
Important Event Handlers	
edit1TB :: focusin ()	
edit2TB :: focusin ()	
noneditCB :: focusin ()	
exitMI :: activate ()	
cutMI :: activate ()	
copyMI :: activate ()	
pasteMI :: activate ()	
deleteMI :: activate ()	
Extension Summary	
Important Extensions	
editWin :: class extension	
editWin :: preheader extension	
editWin :: prebody extension	

NewEra by Example

noneditCB :: focusin ()

editw.win - in noneditCB handler for xCheckBox.focusin event

VARIABLE

win editWin = getWindow()

CALL win.setEditItemsEnabled(SELF)

Works in the same way as the `edit1TB.focusin()`, but passes the FALSE parameter to disable the editing menu items while the user is in the check box.

Calls the `setEditItemsEnabled()` function to disable the editing menu items when the user enters the check box. The user cannot paste into a check box.

Edit window

Text to edit

More text to edit

☒ CheckBox

Use the edit menu to cut and paste text from one textbox to the other. Use the button to start a text editor so you can paste in the editor. Tab to the checkbox and button to see disabling of the menu items.

Fig. 9P

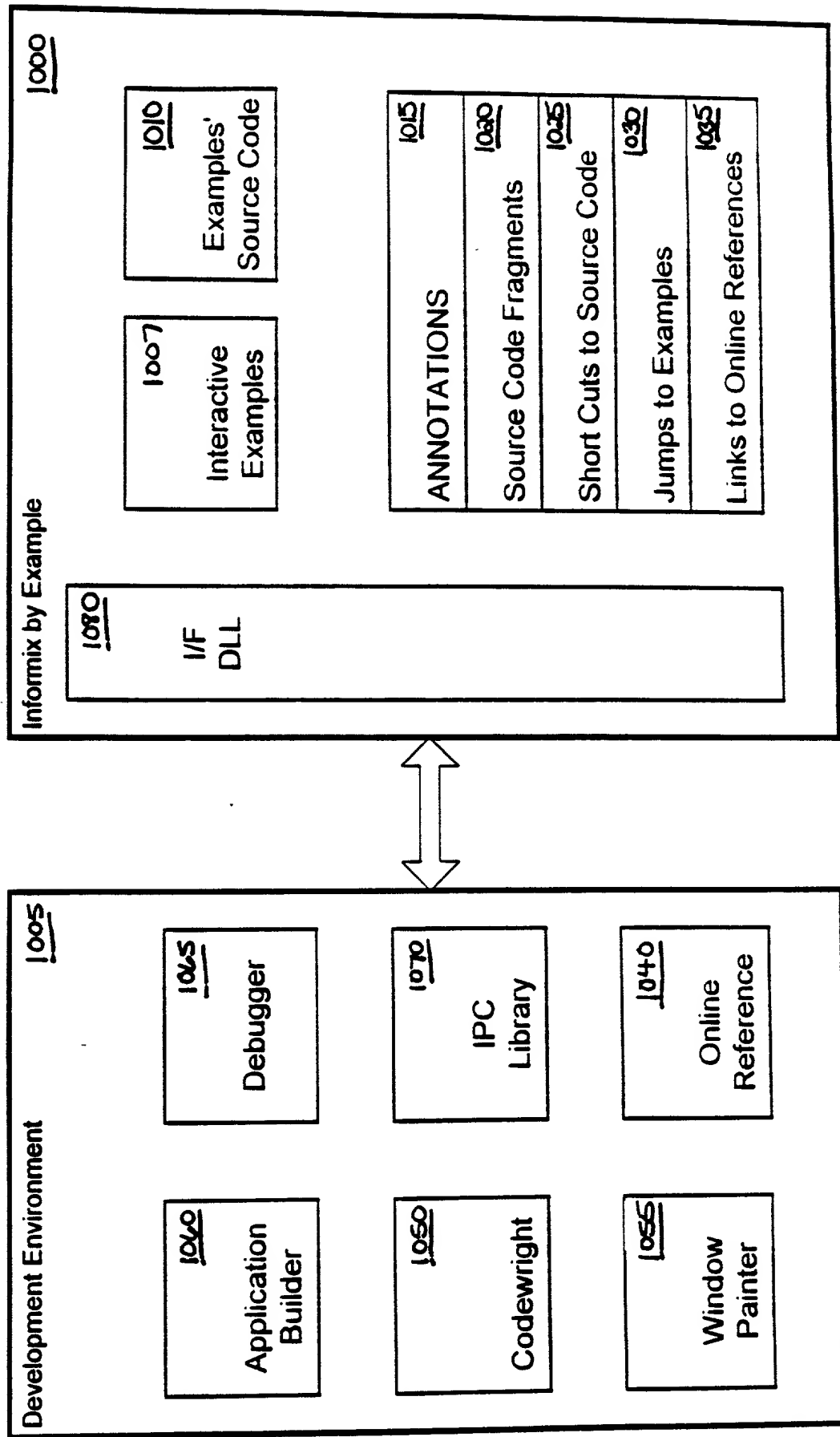


Fig. 10

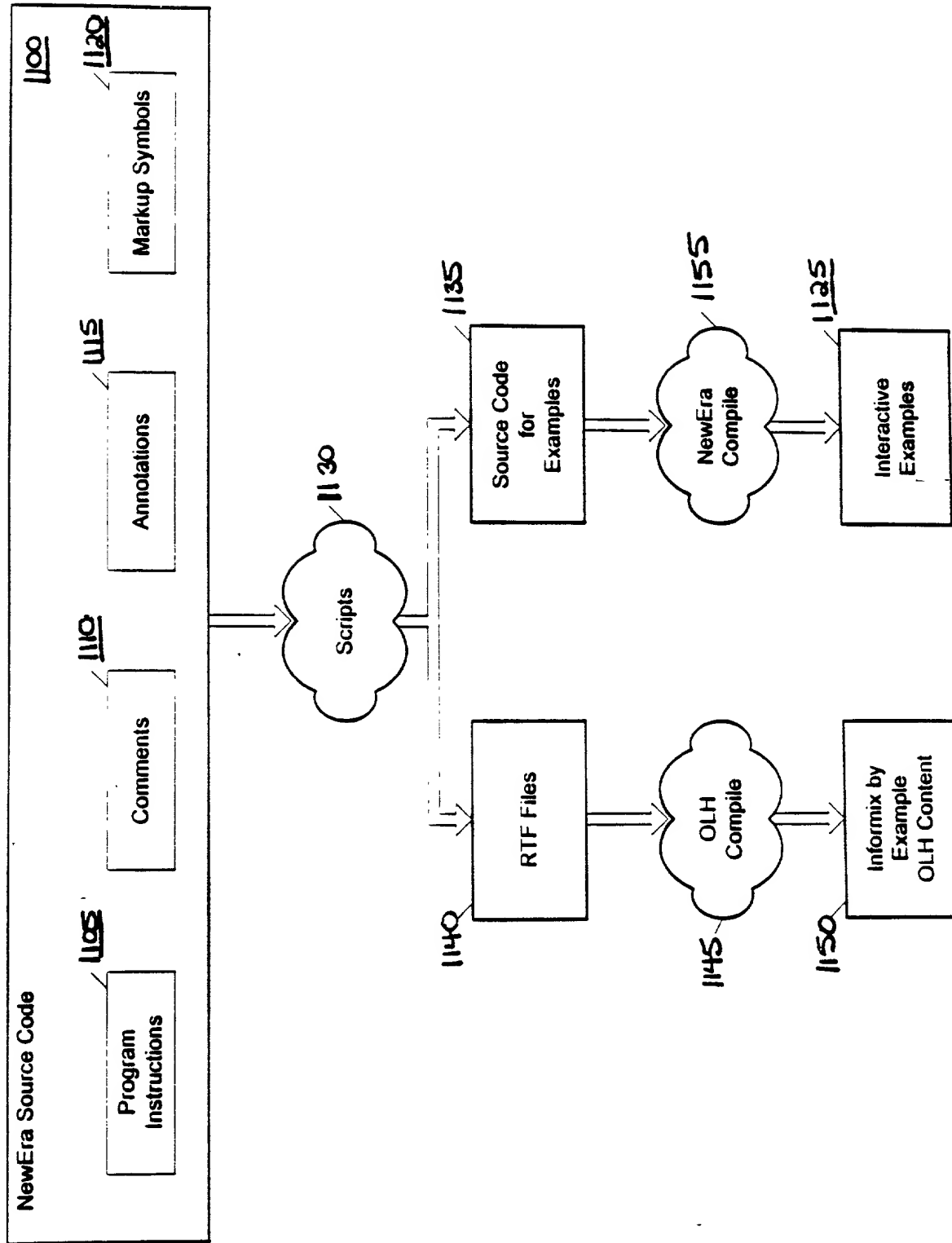


Fig. 11

```

FUNCTION driveStockRpt( destType SMALLINT, destName CHAR(*) ) RETUR
1200 NING VOID
{.normal
Since objects, in particular ixRow objects, cannot be passed
as arguments to the report formatter, rows of fetched data will
be unpacked into a record that matches the data types and lengths
of elements in the fetched rows.
}
VARIABLE
    stockRec RECORD
        mn CHAR(15),      -- manufact.manu_name
        sn SMALLINT,      -- stock.stock_num
        sd CHAR(15),      -- stock.description
        sp MONEY(6,2),    -- stock.unit_price
        su CHAR(4)        -- stock.unit
    END RECORD,

    stockStmt ixSQLStmt,
    stmtString CHAR(*),
    stockRow ixRow,

    errorCode INTEGER,
    logFile ixErrorLog
1205 {.normal
Use the implicit connection object to create an SQL statement
object. The connection object must already be connected to a
database.
Checking the status of the prepare( ) call will confirm this.
1210 }
{.[edit stmt]
LET stockStmt =
1215 ixSQLConnect::getImplicitConnection().createStmtObject()
{.[file stmt]

```

Fig. 12

1300

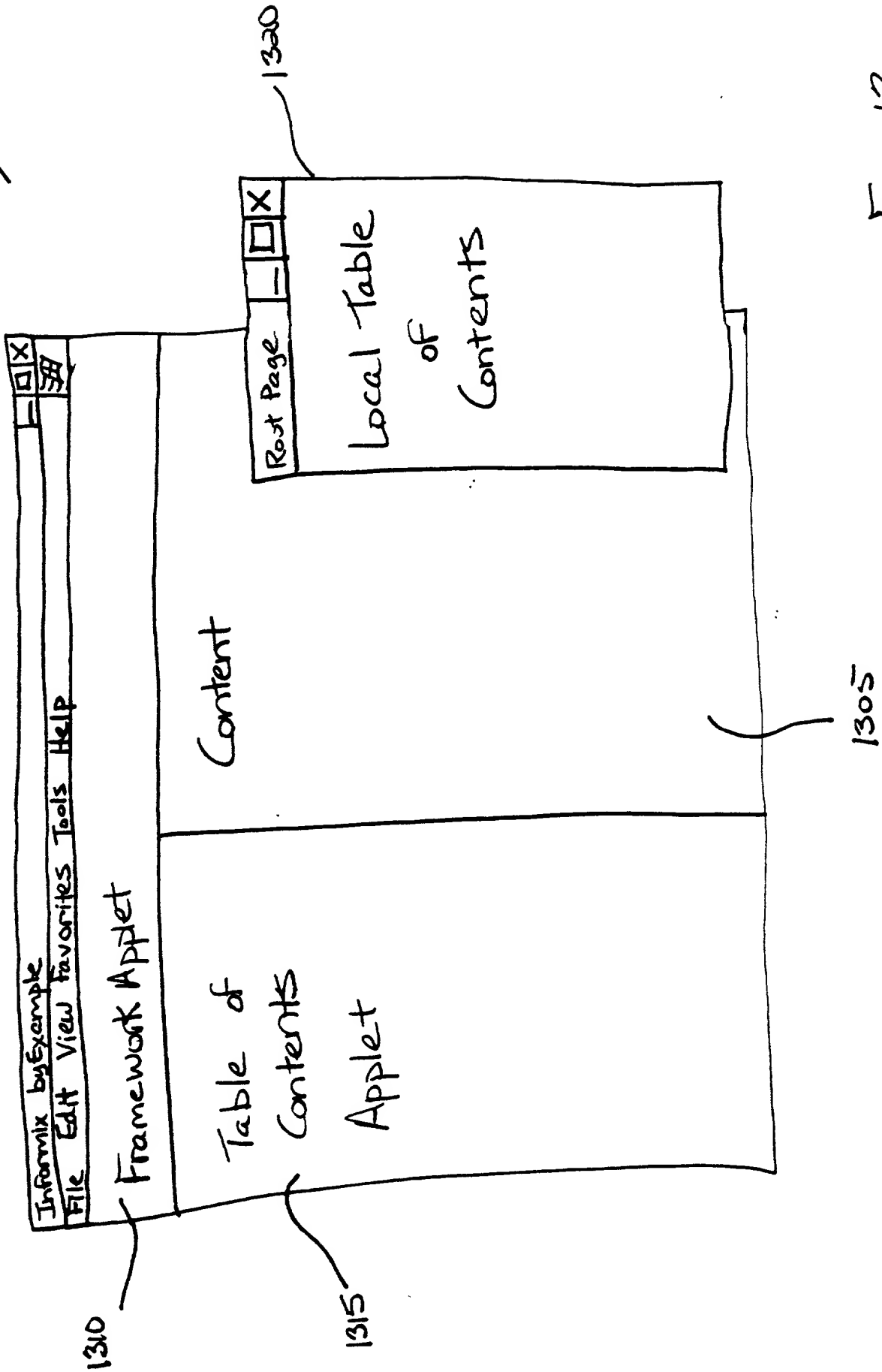


Fig. 13

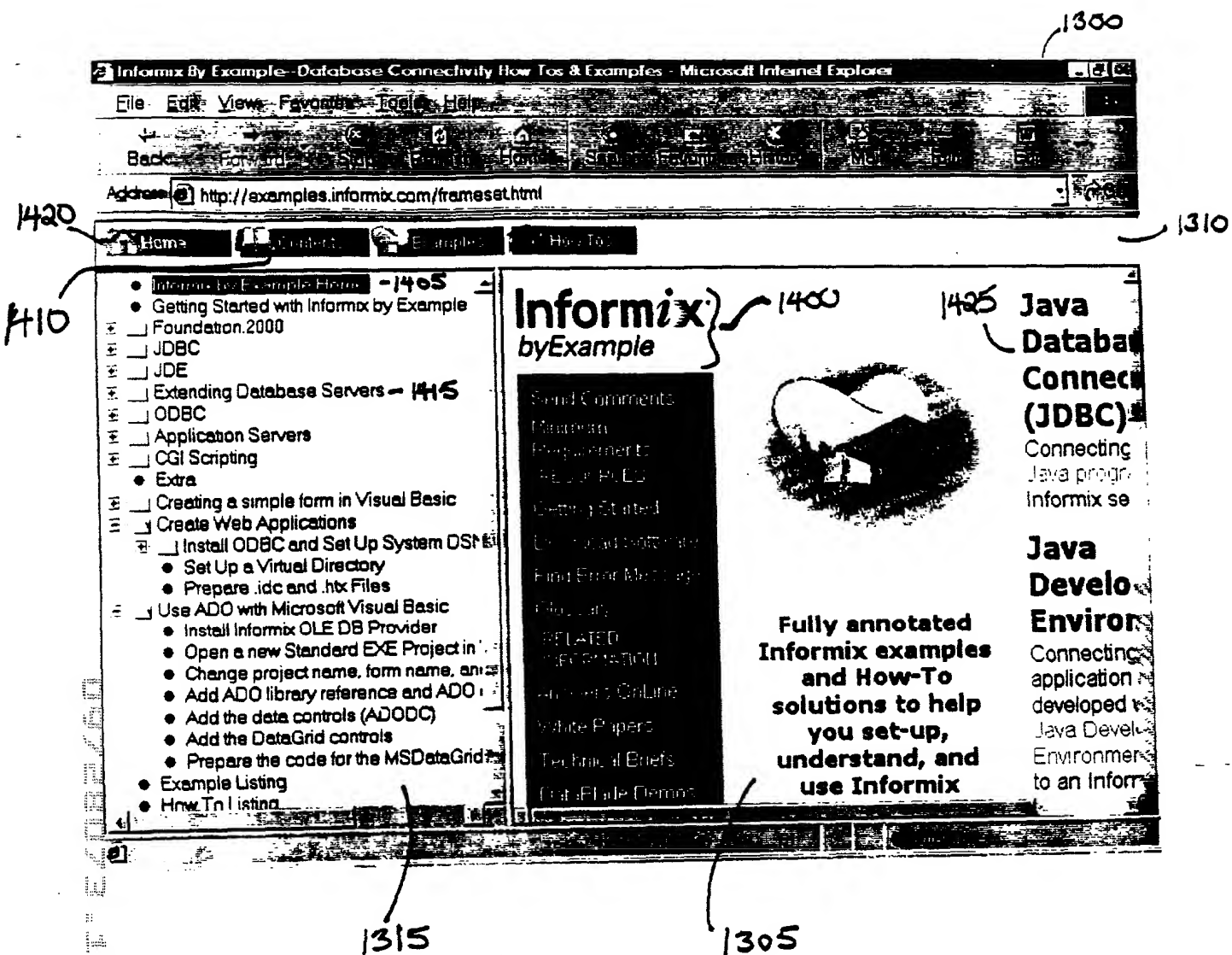


Fig. 14A

Informix[®] byExample

Send Comments
Minimum
Requirements
RELATED
Getting Started
Uninstall Software
Find Error Message
Glossary
RELATED
INFORMATION
Articles OnLine
White Papers
Technical Briefs
DataBlade Demos



1410
**Fully annotated
Informix examples
and How-To
solutions to help
you set-up,
understand, and
use Informix
technology with**

1425 Java Database Connectivity (JDBC)

Connecting your Java program to an Informix server.

Java Development Environment

Connecting a web application developed within a Java Development Environment (JDE) to an Informix database environment.

Extending Database Servers

Using and Writing DataBlade Modules.

Open Database Connectivity (ODBC)

Connecting your C program to a database server.

Application Servers

Using network application servers in conjunction

Fig. 14B

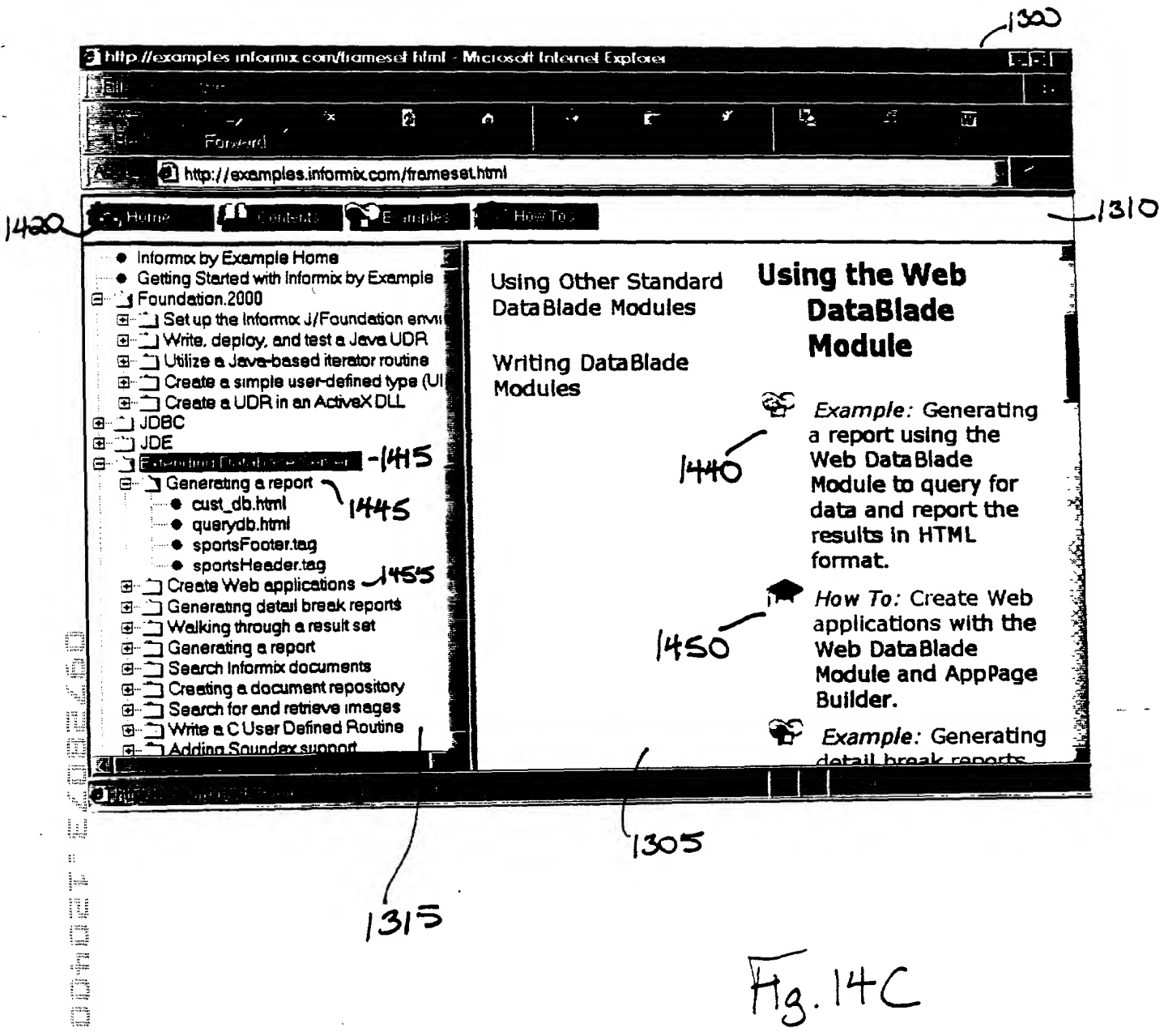


Fig. 14C

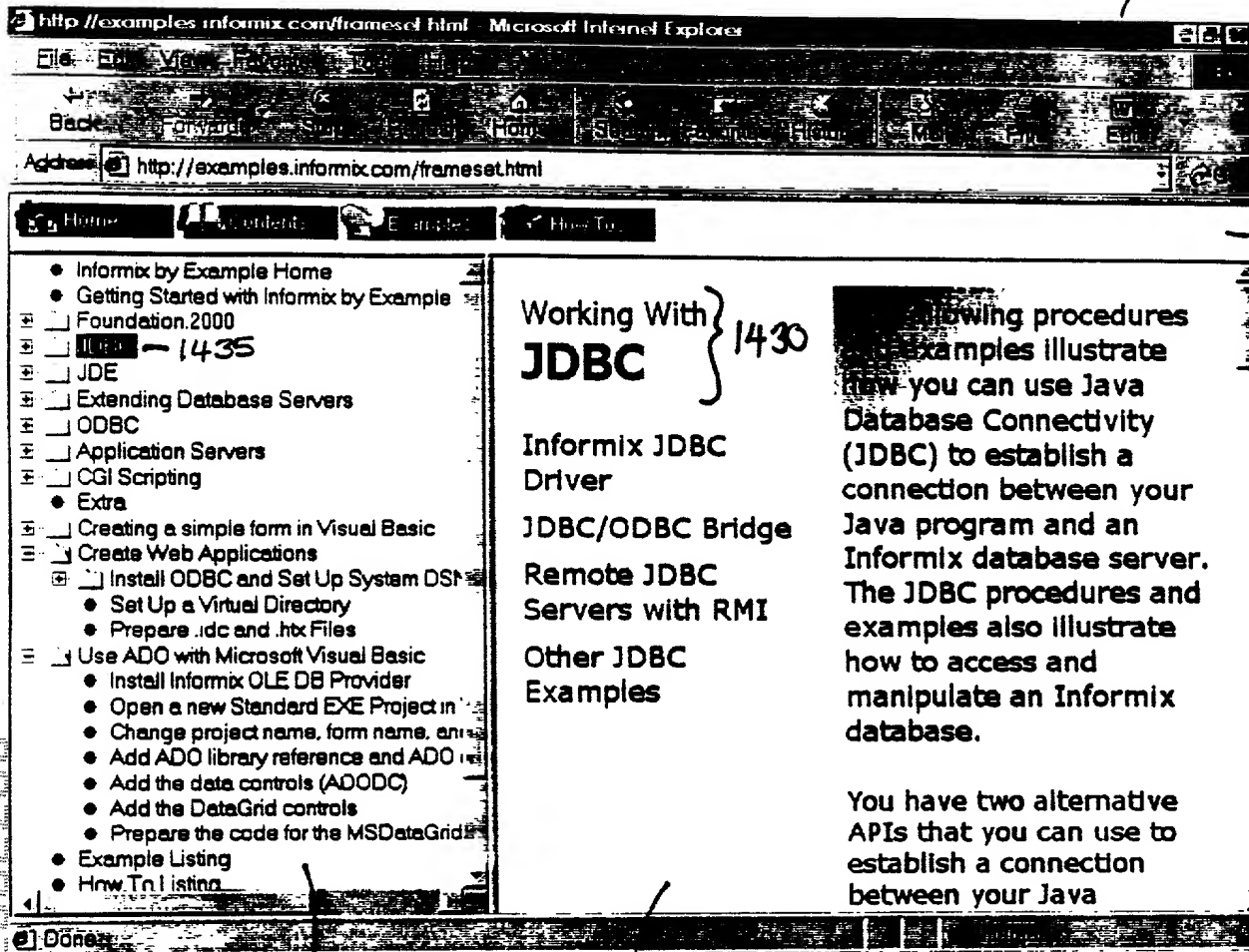


Fig.14D

Generating a Report Using the Web Datablade Module

1500



Contributed by
Erik Hennum,
byExample Team

The customer report example demonstrates how to use the Web DataBlade module to query for data and report the results in HTML format.

cust_db.html ~ 1510

1505

This app page accepts a query and generates an HTML report

querydb.html ~ 1510

1505

This HTML page contains a form that invokes an app page

sportsFooter.tag ~ 1510


1505

The sportsFooter dynamic tag generates the footer for an app page.

sportsHeader.tag ~ 1510

1505

The sportsHeader dynamic tag generates the header for an app page.

 [Click here to view or print all of the source files for this example.](#)

Copyright © 1999 Informix Software. All Rights Reserved.
Terms and Conditions governing the use of this website.

Fig. 15B

DD4027-2032-6B

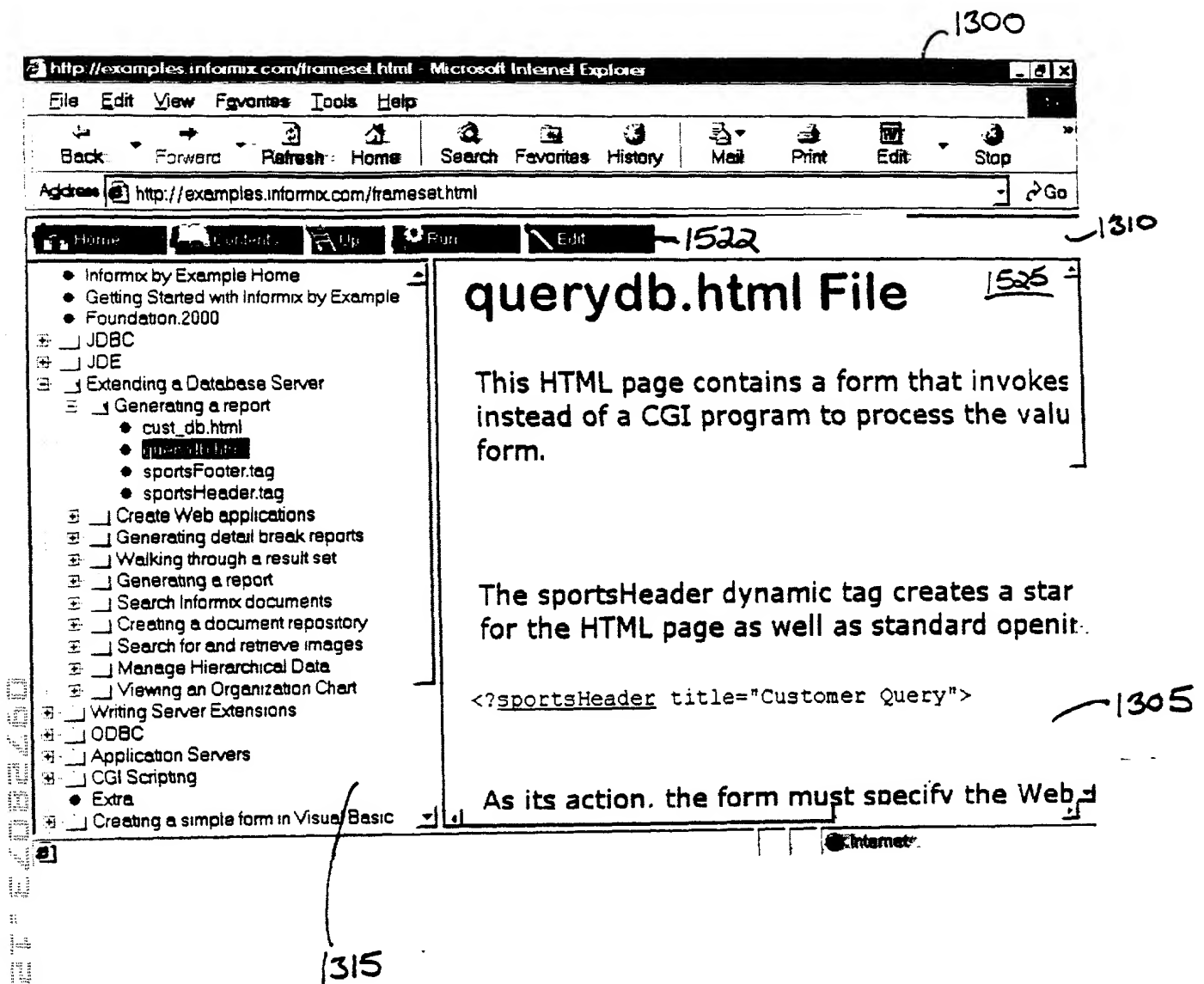


Fig. 15C

querydb.html File

This HTML page contains a form that invokes an app page instead of a CGI program to process the values in the form.

The sportsHeader dynamic tag creates a standard header for the HTML page as well as standard opening text.

<?sportsHeader title="Customer Query">

As its action, the form must specify the Web Driver utility.

<P>
<FORM ACTION="<?MIVAR>\$WEB_HOME<?/MIVAR" METHOD="GET">

To specify the app page, the form must use a hidden input component. The input component must have a name of **Mival** and a value that's the name of the app page. The input component below specifies the cust_db.html app page.

<INPUT TYPE="HIDDEN" NAME="Mival" VALUE="/examples/CustRpt/cust_db.html">

Optional state:

<INPUT TYPE="TEXT" NAME="selectState" SIZE="3" MAXLENGTH="2">

<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">

</FORM>

</P>

</BODY>

</HTML>

Copyright © 1999 Informix Software. All Rights Reserved.

Terms and Conditions governing the use of this website.

Fig. 15D

```

<!-- <ibyx>
<intro>
<p><abstract>This HTML page contains a form that invokes
an app page</abstract> instead of a CGI program to process
the values in the form.
</p>
</intro>
</ibyx> -->

<!-- <ibyx>
<p>The sportsHeader dynamic tag creates a standard header
for the HTML page as well as standard opening text.
</p>
</ibyx> -->
<?sportsHeader title="Customer Query">

<!-- <ibyx>
<p>As its action, the form must specify the Web Driver utility.
</p>
</ibyx> -->
<P>
<FORM ACTION="<?MIVAR>$WEB_HOME</MIVAR>" METHOD="GET">

<!-- <ibyx>
<p>To specify the app page, the form must use a hidden input component.
The input component must have a name of <strong>Mival</strong> and
a value that's the name of the app page. The input component below
specifies the <a href="cust_db.html">cust_db.html</a> app page.
</p>
</ibyx> -->
<INPUT TYPE="HIDDEN" NAME="Mival" VALUE="/examples/CustRpt/cust_db.html">

Optional state:
<INPUT TYPE="TEXT" NAME="selectState" SIZE="3" MAXLENGTH="2">
<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">

</FORM>
</P>

<?annotate>

</BODY>
</HTML>

```

} 1535

} 1535

1560

Fig. 15E

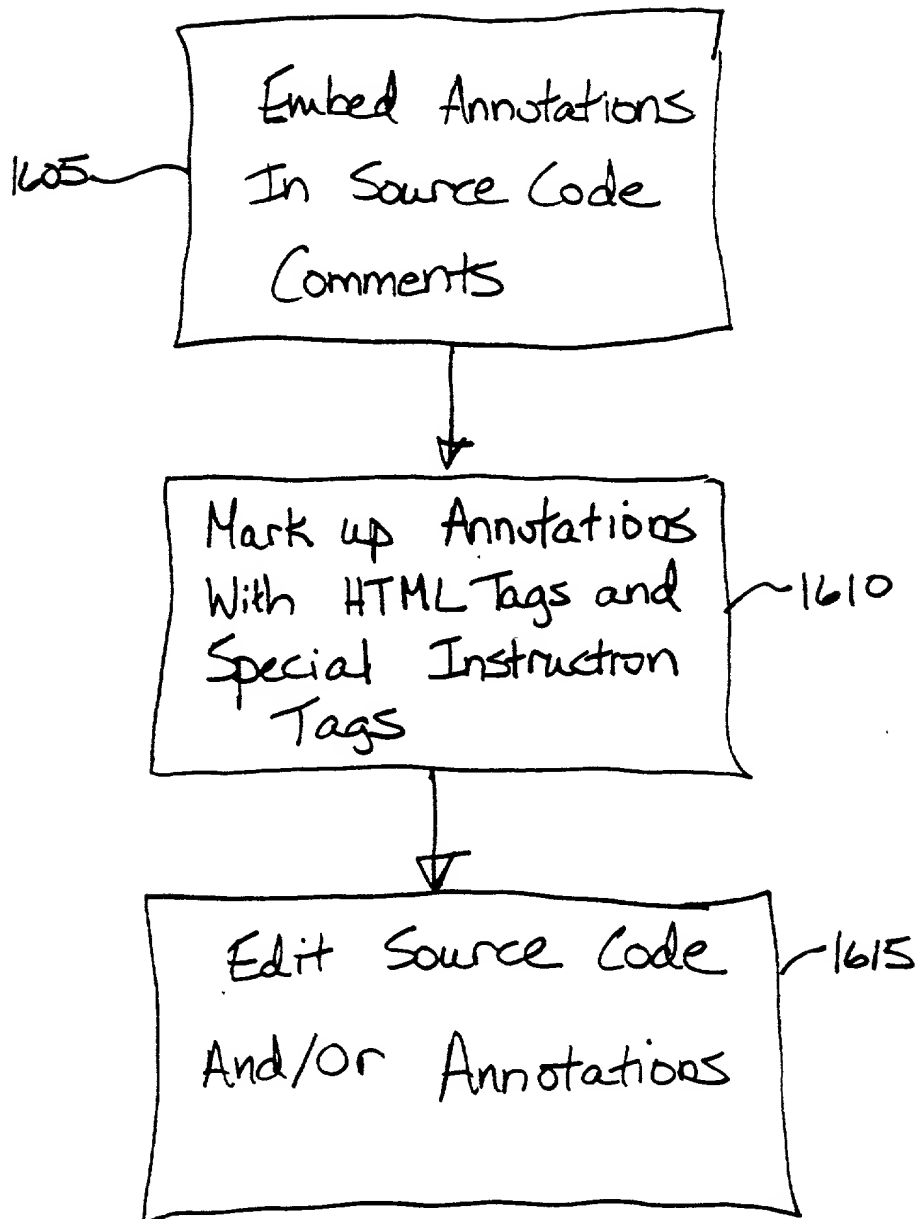


Fig. 16A

1620

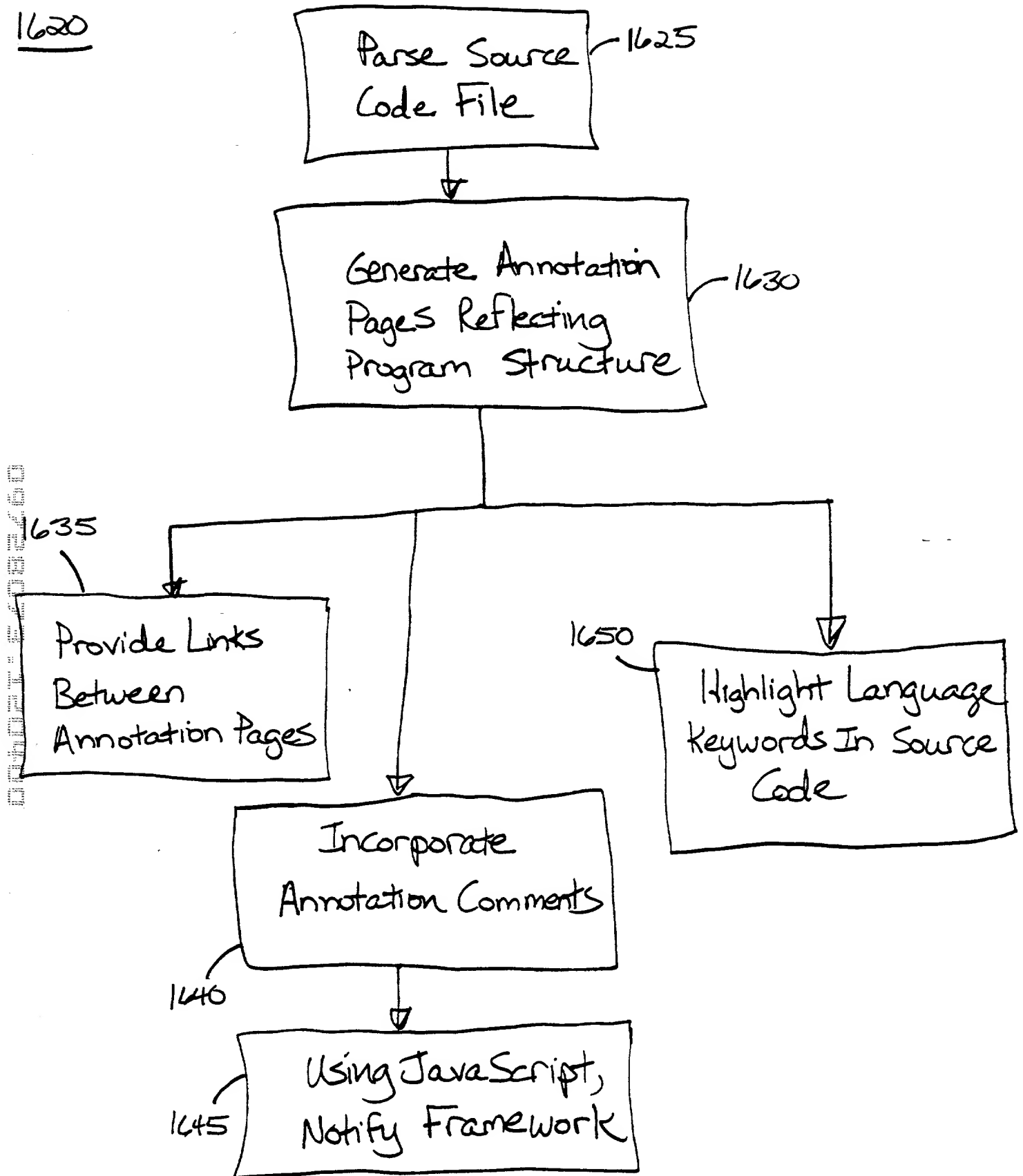


Fig. 16B

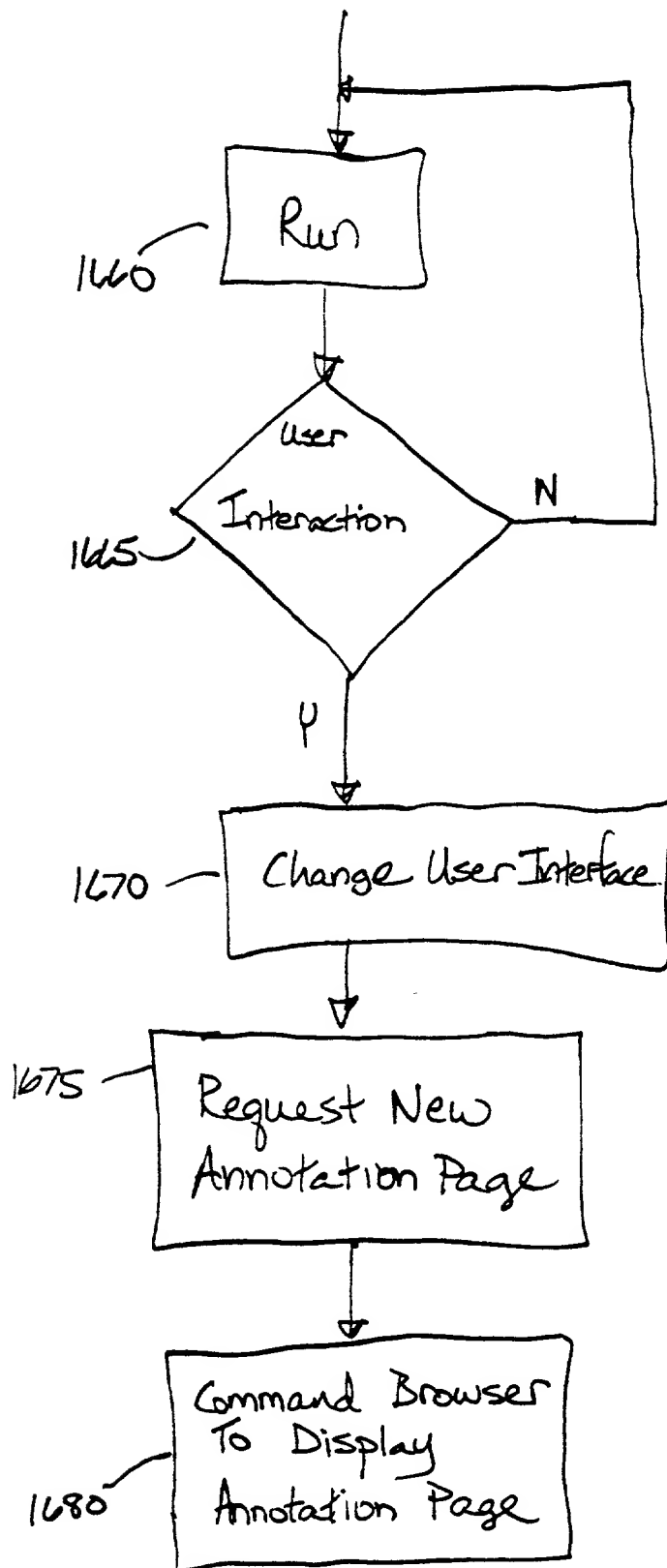


Fig. 16C

004027-2002-260

1300

http://examples.informix.com/frameset.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Refresh Home Search Favorites History Mail Print Edit Stop

Address http://examples.informix.com/frameset.html Go

Home Contents Up Run Edit

1310

- Informix by Example Home
- Getting Started with Informix by Example
- Foundation.2000
- JDBC
- JDE
- Extending a Database Server
 - Generating a report
 - 1700
 - cust_db.html
 - querydb.html
 - sportsFooter tag
 - sportsHeader tag
 - Create Web applications
 - Generating detail break reports
 - Walking through a result set
 - Generating a report
 - Search Informix documents
 - Creating a document repository
 - Search for and retrieve images
 - Manage Hierarchical Data
 - Viewing an Organization Chart
- Writing Server Extensions
- ODBC
- Application Servers
- CGI Scripting
 - Extra
- Creating a simple form in Visual Basic

sportsHeader.tag File

1940

The sportsHeader dynamic tag generates the an app page. This tag ensures that every HT which it is used will have a standard header.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//E
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="te
```

The app page that uses this dynamic tag can title as a parameter. If the title is not supplied following block sets a default title.

Internet

1315 1305

Fig. 17A

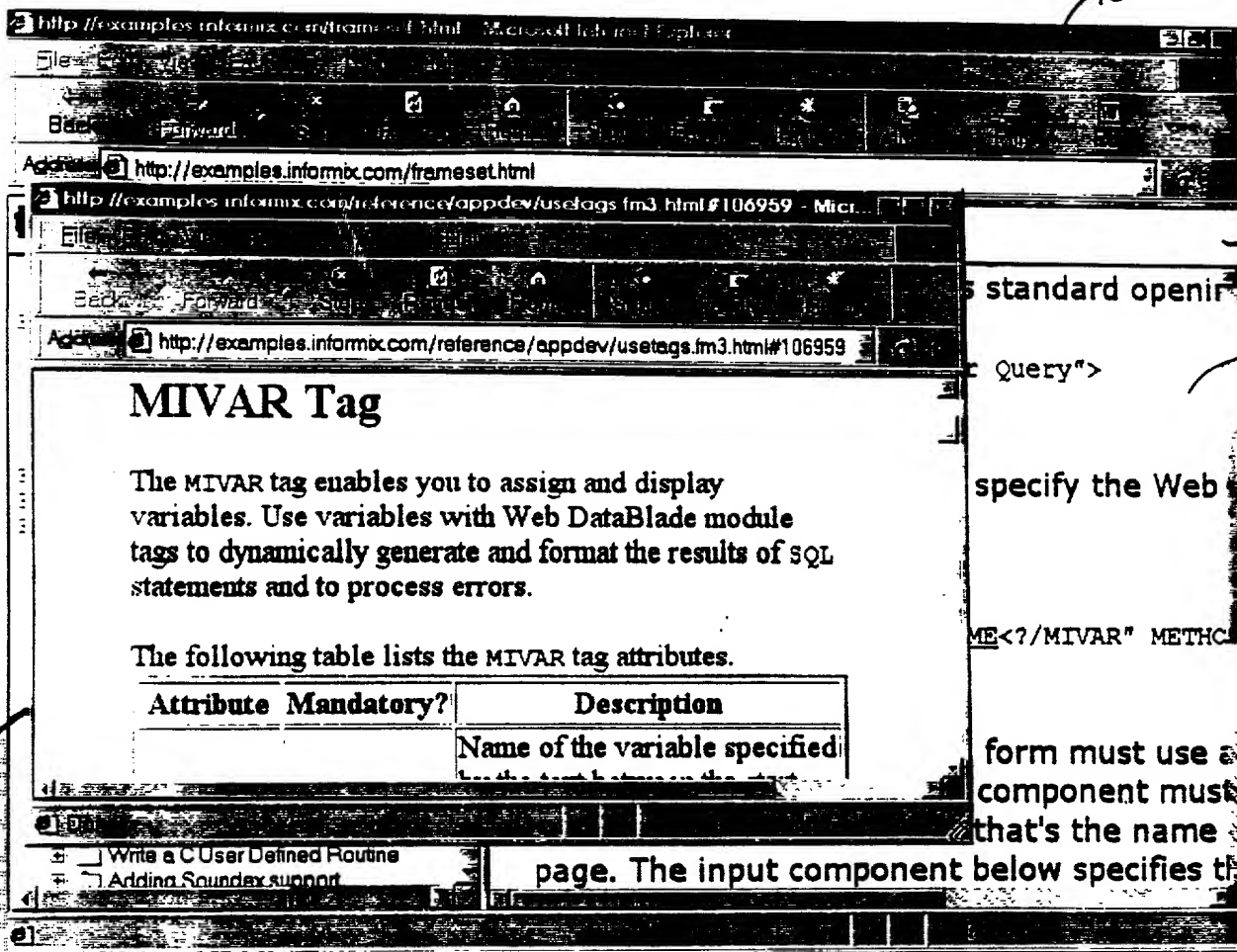


Fig. 17B

1300

1710

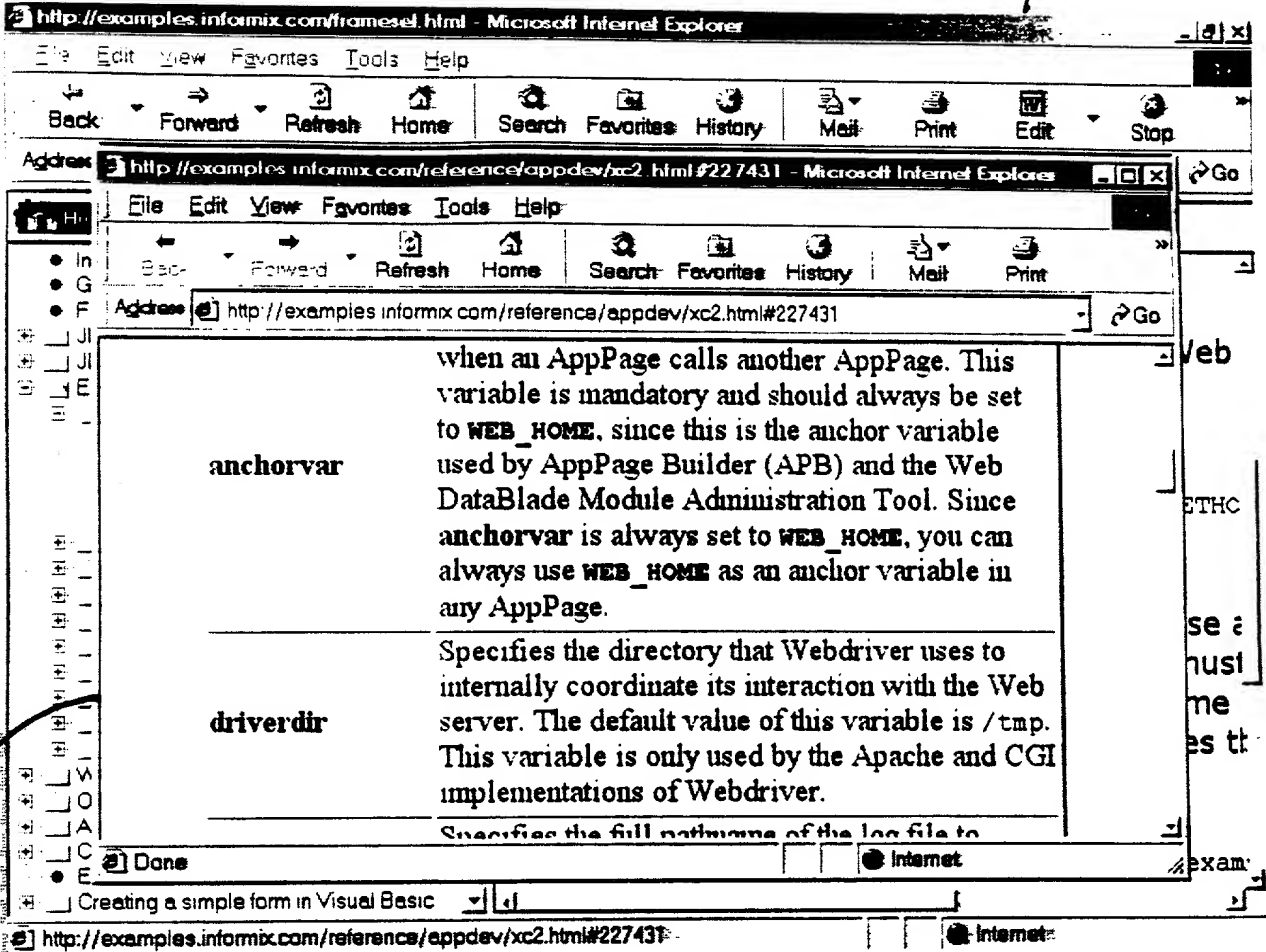


Fig. 17C

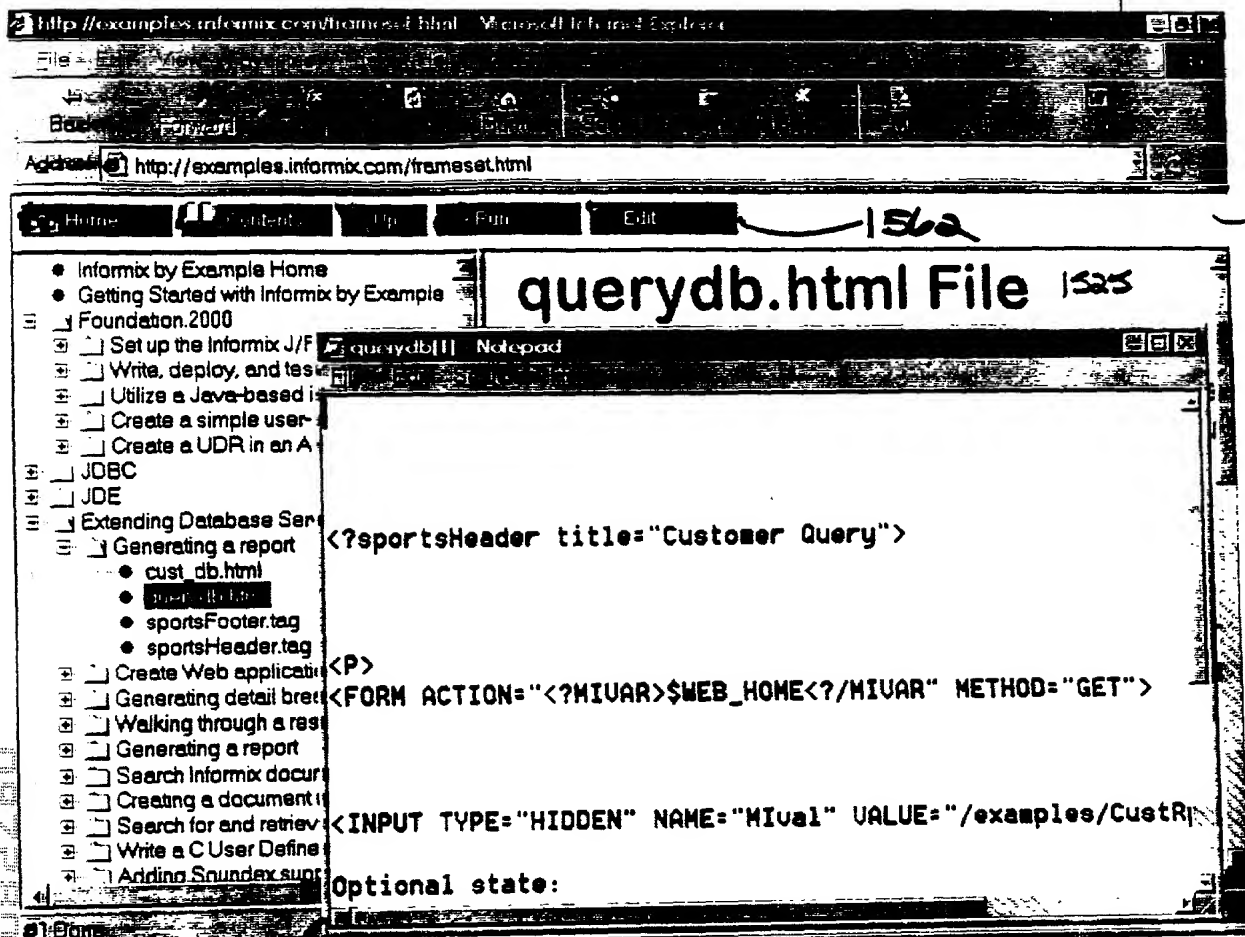
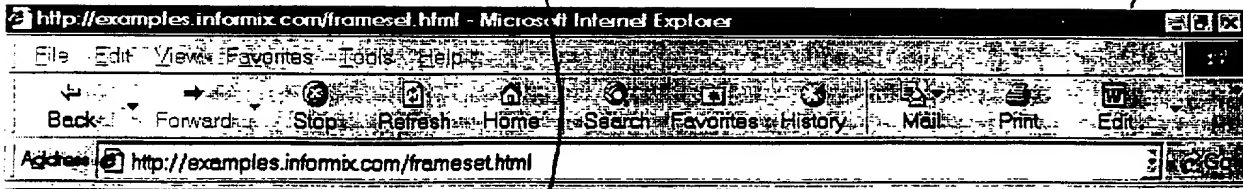


Fig. 17D



- Informix by Example Home
- Getting Started with Informix by Example
 - Foundation.2000
 - Set up the Informix J/Foundation environment
 - Write, deploy, and test a Java UDR
 - Utilize a Java-based iterator routine
 - Create a simple user-defined type (UDT)
 - Create a UDR in an ActiveX DLL
 - JDBC
 - JDE
- Extending Database Servers
 - Generating a report
 - 1580
 - 1700
 - cust_db.html
 - querydb.html
 - sportsFooter.tag
 - sportsHeader.tag
 - Create Web applications
 - Generating detail break reports
 - Walking through a result set
 - Generating a report
 - Search Informix documents
 - Creating a document repository
 - Search for and retrieve images
 - Write a C User Defined Routine
 - Adding Soundex support

querydb.html File 1525

This HT
instead
form.

1700
The spo
for the

1305

<?sportsHeader title="Customer Query">

As its action, the form must specify the Web

Fig. 18A

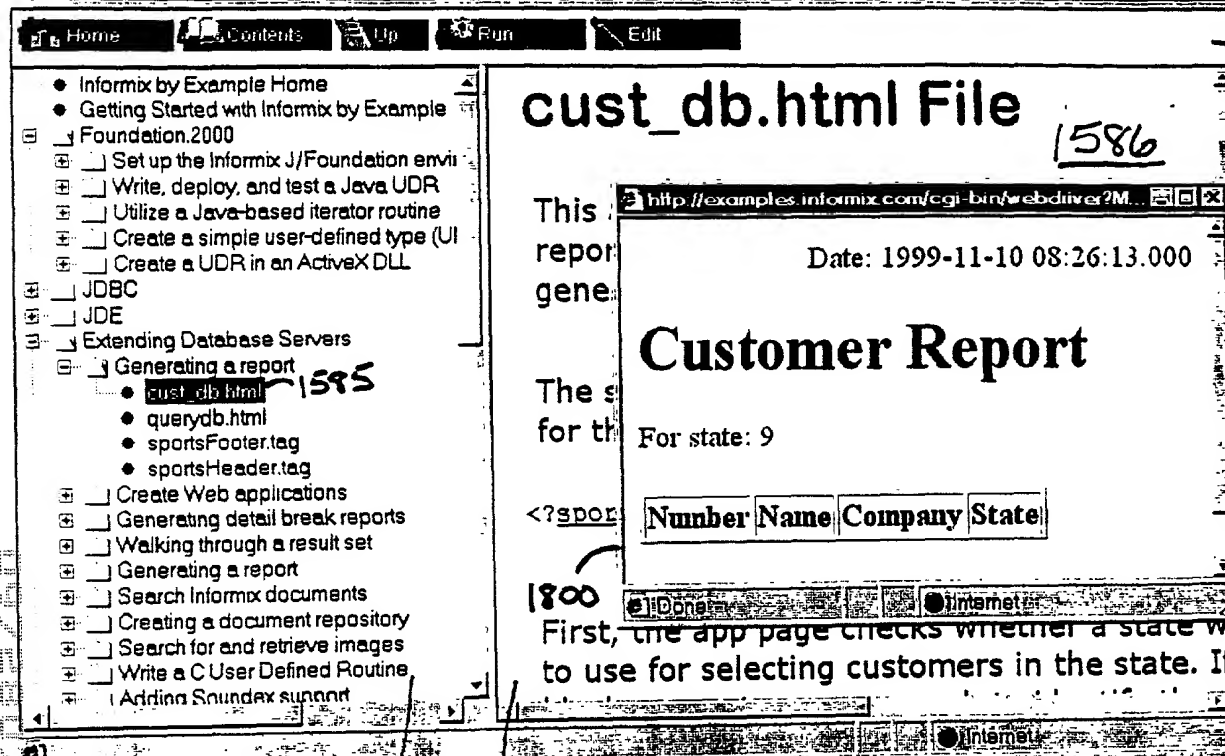


Fig. 18B

```

<!-- <ibyx>
<intro>
<p><abstract>This app page accepts a query and generates an HTML
report</abstract> in response.
The app page uses dynamic tags to generate the header and footer for the
HTML report.
</p>
</intro>
</ibyx> -->
<!-- <ibyx>
<p>The sportsHeader dynamic tag creates a standard header
for the HTML page as well as standard opening text.
</p>
</ibyx> -->
<?sportsHeader title="Customer Report">

<!-- <ibyx>
<p>First, the app page checks whether a state was specified to use for
selecting customers in the state. If so, the block generates a
paragraph to identify the state.
</p>
</ibyx> -->
<?MIVAR NAME=$WHERE_STR><?/MIVAR>
<?MIBLOCK COND="$(AND,$(XST,$selectState),$(,0,$(STRLEN,$selectState)))">
    <?MIVAR NAME=$WHERE_STR>WHERE state="$selectState"<?/MIVAR>
    <?MIVAR><P>For state: $selectState</P><?/MIVAR>
<?/MIBLOCK>

<!-- <ibyx>
<p>Next, the app page starts the table that will contain the data.
</p>
</ibyx> -->
<P><TABLE BORDER="1">
    <TR>
        <TH>Number</TH><TH>Name</TH><TH>Company</TH><TH>State</TH>
    </TR>

<!-- <ibyx>
<p>The MISQL block queries for customers, optionally selecting only customers
from the specified state. Because the contents of the block are generated
for every row of data, a new table row describes each customer.

The &nbsp; HTML entity is a non-breaking space. By putting a non-breaking
space in each column, we force the Web Browser to display the column even
if the value is null.
</p>
</ibyx> -->
<?MISQL SQL="SELECT customer_num, fname, lname, company, state FROM customer $WHERE_STR;">
    <TR>
        <TD>$1&nbsp;</TD><TD>$2&nbsp;</TD><TD>$3&nbsp;</TD><TD>$4&nbsp;</TD><TD>$5&nbsp;</TD>
    </TR>
<?/MISQL>

</TABLE></P>

<!-- <ibyx>
<p>The sportsFooter dynamic tag creates a standard footer
for the HTML page.
</p>
</ibyx> -->
<?sportsFooter>

```

Fig. 18C

1910 1920 1300

http://examples.informix.com/frameset.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Refresh Home Search Favorites History Mail Print Edit Stop


Address http://examples.informix.com/frameset.html Go

Home Contents Up << >> List 1930

- Informix by Example Home 1905
- Getting Started with Informix by Example
- Foundation.2000
- JDBC
- JDE
- Extending a Database Server
 - Generating a report
 - cust_db.html
 - querydb.html
 - sportsFooter.tag
 - sportsHeader.tag
 - Create Web applications 1902
 - Introduction
 - AppPage Overview 1940
 - Prepare Database
 - Prepare Web DataBlade Module
 - Register Web DataBlade Module
 - Create sbspace
 - Install AppPage Builder
 - Set Up AppPage Builder
 - Create Sample AppPage
 - Generating detail break reports
 - Walking through a result set
 - Generating a report
 - Search Informix documents
 - Creating a document repository

1915 1925

Creating Web Applications with the Web DataBlade AppPage 1900



In this How To we'll build a simple Web application using a Web DataBlade Application Page (AppPage). This Web application will access an Informix database.

Requirements: IDS 9.x, the Web DataBlade module, BladeManager, and a web server. BladeManager is provided with IDS 9.x for UNIX. NT users must install BladeManager from the DataBlade Development Kit (DBDK).

These instructions assume you've already installed Informix IDS 9.x and have it running locally.

1315 1305

Internet

Fig. 19A



Creating Web Applications with the Web DataBlade AppPage

1930

1900

In this How To we'll build a simple Web application using a Web DataBlade Application Page (AppPage). This Web application will access an Informix database.

Requirements: IDS 9.x, the Web DataBlade module, BladeManager, and a web server. BladeManager is provided with IDS 9.x for UNIX. NT users must install BladeManager from the DataBlade Development Kit (DBDK).

These instructions assume you've already installed Informix IDS 9.x and have it running locally.

Define a server connection and prepare a sample database.

1. Define a server connection with setnet32 (NT). Create a sample database or use the stores7 demo database.

► Prepare Database.

Prepare the Web DataBlade development environment.

2. Install the Web DataBlade module and BladeManager.

► Prepare Web DataBlade Development Environment.

3. Register the Web DataBlade module in the demo database with BladeManager.

► Register the Web DataBlade.

4. Create a sbspace for smart large objects, like gifs.

► Create Smart Blob Space (sbspace).

5. Install AppPage Builder in your database.

► Install AppPage Builder in Your Database.

6. Setup AppPage Builder on your web server.

► Setup AppPage Builder on Your Web Server.

7. Create a sample AppPage.

► Create Sample AppPage.

Run the sample application.

8. Enter the URL `http://your_server/scripts/webdriver.exe`.



This How To has been compiled into two separate files for ease of printing. The basic file contains all of the steps you need to Create Web Applications with AppPage Builder. The secondary file contains additional detailed instructions for setting and testing database environment properties.

1300

http://examples.informix.com/frameset.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Refresh Home Search Favorites History Mail

Address http://examples.informix.com/frameset.html

Home Contents Up << < > >> List 1930

- Informix by Example Home
- Getting Started with Informix by Example
- Foundation.2000
- JDBC
- JDE
- Extending a Database Server
 - Generating a report
 - cust_db.html
 - querydb.html
 - sportsFooter.tag
 - sportsHeader.tag
 - Create Web application:
 - Introduction
 - AppPage Overview
 - Prepare Database
 - Prepare Web DataBlade Module
 - Register Web DataBlade Module
 - Create sbpace
 - Install AppPage Builder
 - Set Up AppPage Builder
 - Create Sample AppPage
 - Generating detail break reports
 - Walking through a result set
 - Generating a report
 - Search Informix documents
 - Creating a document repository

Creating Application the Web AppPage

1320

In this How To we'll build a simple Web application using a Web DataBlade Application Page (AppPage). This Web application will access an Informix database.

Requirements: IDS 9.x, the Web DataBlade module, BladeManager, and a web server. BladeManager is provided with IDS 9.x for UNIX. NT users must install BladeManager from the DataBlade Development Kit (DBDK).

These instructions assume you've already installed Informix IDS 9.x and have it running locally.

Warning: Applet Window

Done Internet

1315

1305

Fig. 19C

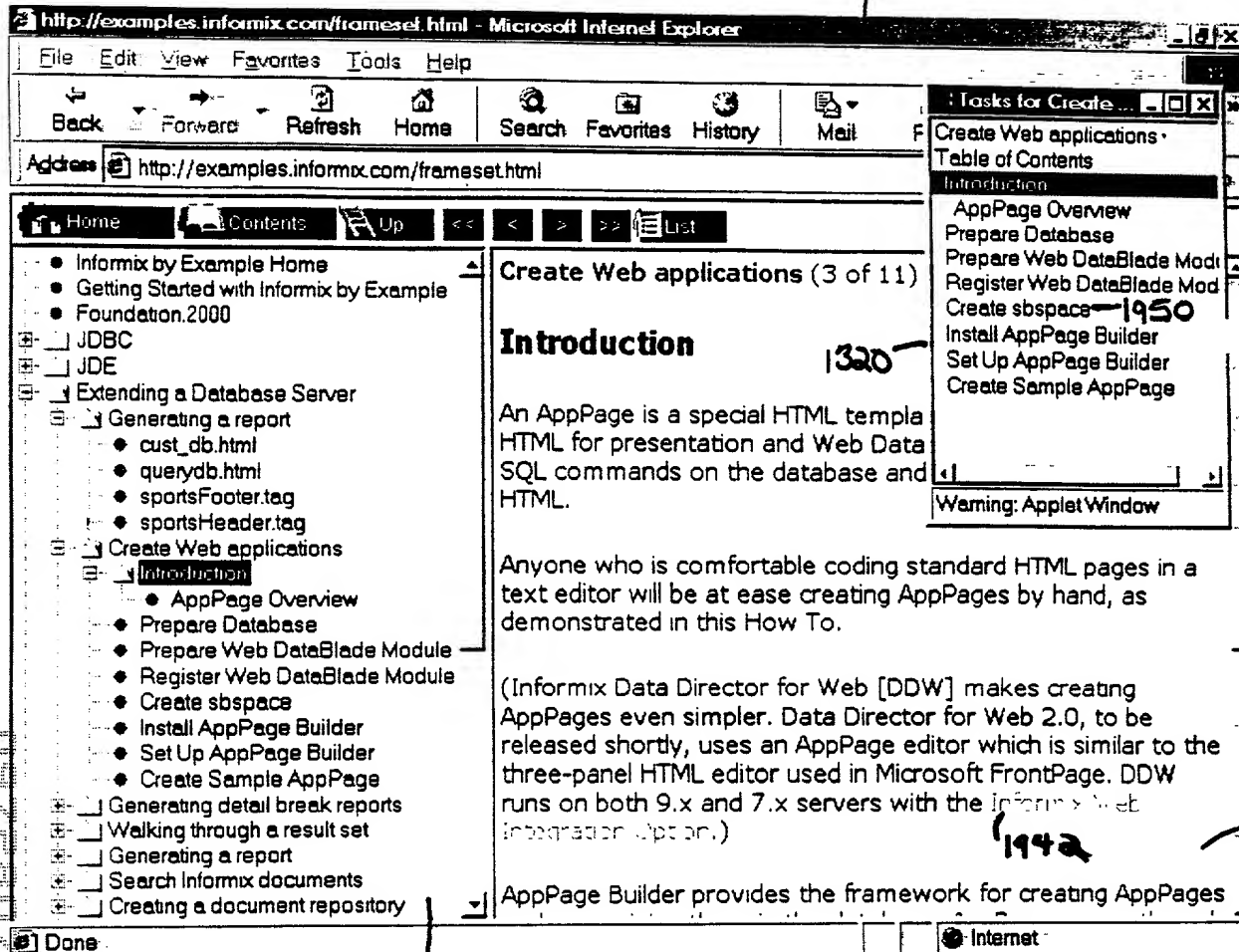


Fig. 19D

1300

http://www.informix.com/wio/ - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Refresh Home Search Favorites History Mail Print Edit Stop

Address http://www.informix.com/wio/ Go

Informix

The one with the smartest data wins™

1945

Home | Contact Us | Search | Corporate | Solutions | **Products** | Services | Partners

PRODUCTS

- Packages & Solutions
- Servers
- Integration Products
- Tools
- Technologies
- All Informix Products

RELATED INFORMATION

- Year 2000
- Online Documentation
- Try & Buy
- Success Stories

Informix Web Integration Option Overview

Informix Web Integration Option provides high-performance connectivity between Web servers and Informix Dynamic Server. Web Integration Option enables Web developers to rapidly create, manage, and deploy value-added Web applications that dynamically deliver tailored Web pages to a corporation's Internet, intranet, and extranet users.

With its openness and highly optimized integration, Web Integration Option enables organizations to exploit the power of Informix Dynamic Server while using the tools and languages they already know.

Web Integration Option offers:

Internet

1305

Fig. 19E

1300

http://examples.informix.com/frameset.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Refresh Home Search Favorites History Mail Print Edit

Address http://examples.informix.com/frameset.html

Home Contents Up << < > >> List

- Informix by Example Home
- Getting Started with Informix by Example
- Foundation.2000
- JDBC
- JDE
- Extending a Database Server
 - Generating a report
 - cust_db.html
 - querydb.html
 - sportsFooter.tag
 - sportsHeader.tag
 - Create Web applications
 - Introduction
 - AppPage Overview
 - Prepare Database
 - Prepare Web DataBlade Module
 - Register Web DataBlade Module
 - Create sbpace
 - Install AppPage Builder
 - Set Up AppPage Builder
 - Create Sample AppPage
 - Generating detail break reports
 - Walking through a result set
 - Generating a report
 - Search Informix documents
 - Creating a document repository

Create Web applications

Table of Contents

1. Web DataBlade / AppP
2. Introduction
 1. AppPage Overview
3. Prepare Database
4. Prepare Web DataBlade Module
5. Register Web DataBlade Module
6. Create sbpace
7. Install AppPage Builder
8. Set Up AppPage Builder
9. Create Sample AppPage

Warning: Applet Window

Internet

1315

1320

1305

Fig. 19F

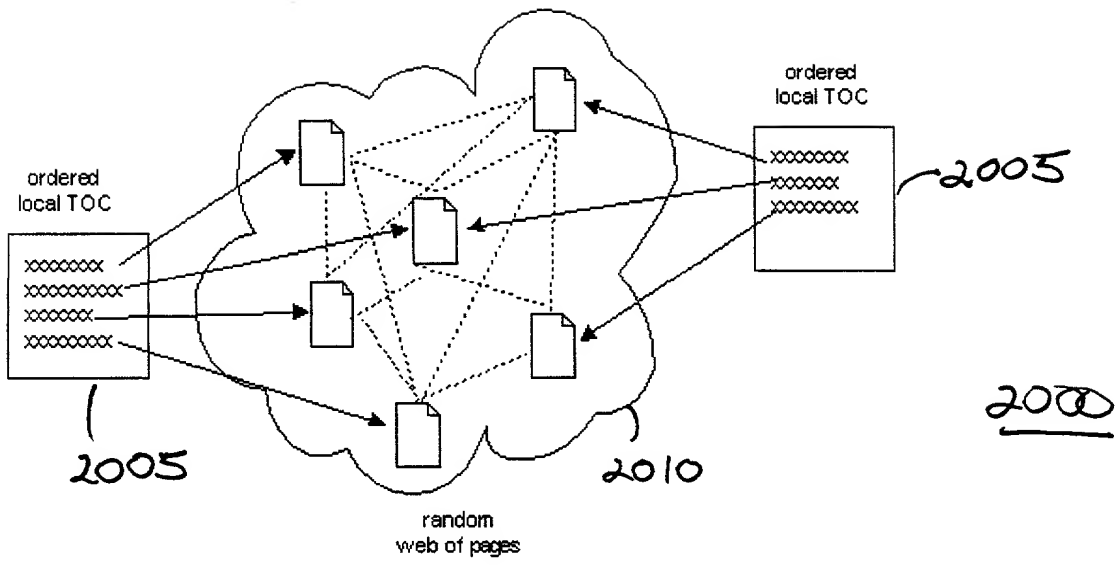


Fig. 20

DocId: 6203260